Technische Universität München
TUM School of Medicine and Health

Identification, characterization and validation of neoantigens and neoantigen-reactive T cells in their distinct tumor microenvironment of patients included in the ImmuNEO MASTER pan-cancer cohort

Celina Tretter

Vollständiger Abdruck der von der TUM School of Medicine and Health der Technischen Universität München zur Erlangung einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Thomas Korn

Prüfende der Dissertation:

1. Prof. Dr. Angela Krackhardt
2. Prof. Dr. Julien Gagneur

Die Dissertation wurde am 10.01.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Medicine and Health am 10.04.2024 angenommen.

Fakultät für Medizin

Klinik und Poliklinik für Innere Medizin III, Hämatologie und Onkologie,
Klinikum rechts der Isar der Technischen Universität München

Identification, characterization and validation of neoantigens and neoantigen-reactive T cells in their distinct tumor microenvironment of patients included in the ImmuNEO MASTER pan-cancer cohort

Celina Tretter

Parts of this thesis have already been published:

Celina Tretter*, Niklas de Andrade Krätzig*, Matteo Pecoraro, Sebastian Lange, Philipp Seifert, Clara von Frankenberg, Johannes Untch, Gabriela Zuleger, Mathias Wilhelm, Daniel P Zolg, Florian S Dreyer, Eva Bräunlein, Thomas Engleitner, Sebastian Uhrig, Melanie Boxberg, Katja Steiger, Julia Slotta-Huspenina, Sebastian Ochsenreither, Nikolas von Bubnoff, Sebastian Bauer, Melanie Boerries, Philipp J Jost, Kristina Schenck, Iska Dresing, Florian Bassermann, Helmut Friess, Daniel Reim, Konrad Grützmann, Katrin Pfütze, Barbara Klink, Evelin Schrock, Bernhard Haller, Bernhard Kuster, Matthias Mann, Wilko Weichert, Stefan Fröhling, Roland Rad, Michael Hiltensperger & Angela M Krackhardt, Proteogenomic analysis reveals RNA as a source for tumor-agnostic neoantigen identification. Nat Communications, 14, 4632 (2023). https://doi.org/10.1038/s41467-023-39570-7.
(*equal contribution)
Detailed information on contributions of authors and cooperation partners to this thesis can be found in section 6.7

Table of content

Abbreviations

%	percentage
°C	degree Celsius
7AAD	7-amino-actinomycin D
A	adenosine
Aa	amino acid
acDC	accelerated cocultured DC assay
ADAR	Adenosine Deaminases Acting on RNA
ALL	acute lymphocytic leukemia
AML	acute myeloid leukemia
APC(s)	antigen-presenting cell(s)
bp	base pairs
BSA	bovine serum albumine
Ca	carcinoma
CAR	chimeric antigen receptor
CD	cluster of differentiation
CDS	coding sequence
CEA	carcinoembryonic antigen
CML	chronic myeloid leukemia
CNV	copy number variation
CMV	cytomegalovirus
CPTAC	Clinical Proteomic Tumor Analysis Consortium
CTA	cancer testis antigen
CTLA-4	cytotoxic T-lymphocyte-associated protein 4
Da	Dalton
DAI	differential agretopicity index
DC(s)	dendritic cell(s)
DKFZ	German Cancer Research Centre
DKTK	German consortium for translational cancer research
DLBCL	diffuse large B cell lymphoma
DMSO	dimethyl sulfoxide
dMMR	mis-match-repair-deficient
DSRCT	desmoplastic small round cell tumor
E:T	effecter-to-target ratio
EBV	Epstein Barr virus

ELISA	enzyme-linked immunosorbent assay
ELISpot	enzyme-linked immunospot assay
EMA	ethidium monoazide
EpCAM	epithelial cell adhesion molecule
FACS	fluorescence activated cell sorting
FCS	fetal calf serum
FCS-A/W/H	forward scatter-area/width/height
FDA	U.S. Food and Drug Administration
FDR	false discovery rate
FFPE	formalin-fixed and paraffin-embedded
g	gram
g	gravitational acceleration
G	guanosine
GFP	green fluorescent protein
GIST	gastrointestinal stromal tumor
GM-CSF	granulocyte-macrophage colony-stimulating factor
GSEA	gene set enrichment analysis
GTEx	Genotype-Tissue Expression portal
Gy	gray
h	hour
HCC	hepatocellular carcinoma
HD	healthy donor
HLA	human leukocyte antigen
HNSCC	head and neck squamous cell cancer
HPLC	high performance liquid chromatography
HPV	human papilloma virus
HRP	horseradish peroxidase
HS	human serum
I	inosine
ICI	immune checkpoint inhibitor/inhibition
IFN-γ	interferon γ
IL	interleukin
IME	immune microenvironment
IN	ImmuNEO
InDel(s)	Insertion and deletion(s)

IP	immunoprecipitation
l	liter
LAG-3	lymphocyte-activation gene 3
LCL	lymphoblastoid cell lines
LN(s)	lymph node(s)
LncRNA(s)	long non-coding RNA(s)
mAb	monoclonal antibody
Mb	megabase
mCRC	metastatic colorectal cancer
MHC	major histocompatibility complex
min	minute
min.	minimum
Mio	million (10^6)
miRNA	micro RNA
ml	milliliter (10^{-3} l)
MPNST	malignant peripheral nerve sheath tumor
MRI	magnetic resonance imaging
MS	mass spectrometry
MSI-H	microsatellite instability-high
NEAA	Non-essential amio acids
ng	nanogram (10^{-9} g)
NGS	next generation sequencing
NK cells	natural killer cells
NKT cells	natural killer T cells
nM	nanomolar (10^{-9} M)
NSCLC	non-small-cell lung cancer
NTRK	neurotrophic receptor tyrosine kinase
ORF	open reading frame
p.p.m	parts per million
PBMC	Peripheral blood mononuclear cells
PBS	phosphate-buffered saline
PD-1	Programmed cell death protein 1
PD-L1	Programmed cell death protein 1 ligand
PFA	paraformaldehyde
pg	picogram (10^{-12} g)

pH	power of hydrogen, a measure of hydrogen ion concentration
pHLA	peptide-HLA complex
PMBCL	primary mediastinal large B-cell lymphoma
PSM	peptide-spectrum match
RCC	renal cell carcinoma
rh	recombinant human
RNA-seq	RNA sequencing
ROC	receiver operating characteristic
RT	room temperature
RT	retention time
SA	spectral contrast angle
SCCHN	squamous cell cancer of head and neck
SCLC	small-cell lung cancer
SFU	spot forming units
SNP	short nucleotide polymorphisms
SNV(s)	single nucleotide variation(s)
SSC-A/W/H	side scatter-area/width/height
TAA(s)	tumor associated antigen(s)
TCGA	The Cancer Genome Atlas
TCM	T cell medium
Tcm	central memory T cells
TCR	T cell receptor
Tem	effector memory T cells
TFN-a	tumor necrosis factor α
Th	T helper cell
TILs	tumor infiltrating lymphocytes
TIM-3	T cell immunoglobulin and mucin domain-containing protein 3
TLR	toll-like receptor
TMB	tumor mutational burden
TME	tumor microenvironment
Tn	naïve T cells
Treg	regulatory T cell
Trm	resident memory T cell
TSA(s)	tumor specific antigen(s)
U	units

UTR	untranslated region
WES	whole exome sequencing
WGS	whole genome sequencing
wt	wild type
µg	microgram (10^{-6} g)
µl	microliter (10^{-6} l)
µm	micrometer (10^{-6} m)
µM	micromolar (10^{-6} M)

Summary

Multi-omics pan cancer studies have changed the understanding and treatment approach for cancer patients within the past years. Immunotherapy as a systemic therapy approach has shown great potential in pan cancer application, however biomarkers for therapy selection and response prediction are limited and sometimes controversial. Also, the correct target selection for immunotherapies relying on specific tumor antigen identification, such as mRNA vaccines and transgenic T cell therapy, is of great importance. Using proteogenomics approaches for neoantigen identification has already proven successful in specific cancer entities for this purpose.

Within this multi-omics pan-cancer study, 32 patients across 25 tumor types were analyzed by combining proteogenomics for neoantigen identification and characterization with phenotypic and functional analyses for biomarker discovery. A previously established proteogenomic neoantigen identification pipeline was expanded and optimized, that combines deep DNA and RNA sequencing with MS-based immunopeptidomics, followed by immunogenicity assessment of neoantigen candidates. Furthermore, an in-depth validation process that includes a peptide verification as well as tumor-specificity validation of all variants was established.

Thereby, within this work a broad variety of non-self HLA-I-binding peptides were detected in the majority of patients irrespective of tumor entity, of which 32 neoantigen candidates were validated. For eight of these 32 validated neoantigen candidates, immunogenicity was demonstrated in autologous PBMCs and TILs as well as PBMCs of allogenic HLA-matched healthy donors. Most of the neoantigens, both total and immunogenic, can be traced back to variants detected in the RNA dataset. This highlights the significance of RNA as a relatively overlooked reservoir of cancer antigens. Additionally, positive correlation between the quantity of these neoantigens primarily derived from RNA and the presence of CD3+ tumor-infiltrating T cells was observed.

This thesis highlights the importance of RNA-derived variant detection for identifying potentially relevant neoantigen candidates and gives suggestions for improving future neoantigen identification pipelines and neoantigen prioritization for potential clinical application.

Zusammenfassung

Multi-omics pan-cancer Studien haben in den letzten Jahren das Verständnis und den Behandlungsansatz für Krebspatienten stark verändert. Speziell die Immuntherapie als systemischer Therapieansatz hat großes Potenzial für die Anwendung bei Krebserkrankungen unabhängig der Entität gezeigt. Allerdings sind aussagekräftige Biomarker für die Therapieauswahl und die Vorhersage des Therapieansprechens weiterhin begrenzt und manchmal sogar widersprüchlich. Auch die richtige Traget-Auswahl für Immuntherapien, die auf der Identifizierung spezifischer Tumorantigene basieren, wie z. B. mRNA-Impfstoffe und transgene T-Zelltherapie, ist von großer Bedeutung für den Therapieerfolg. Der Einsatz proteogenomischer Ansätze zur Identifizierung von Neoantigenen hat sich hierfür bereits als erfolgreich erwiesen, allerdings meist für spezifische Krebstypen.

In dieser multi-omics pan-cancer Studie wurden 32 Patienten mit 25 Tumortypen analysiert, um mithilfe der Proteogenomik zum einen Neoantigene zu identifizieren und zu charakterisieren, und zum anderen mögliche Biomarker durch phänotypische und funktionelle Analysen zu entdecken. Hierfür wurde eine proteogenomische Neoantigen-Identifizierungspipeline erweitert und optimiert, in der DNA- und RNA-Sequenzierung mit MS-basierter Immunpeptidomik kombiniert werden, gefolgt von einem Immunogenitätstest der so identifizierten Neoantigen-Kandidaten. Darüber hinaus wurde ein Validierungsprozess etabliert, durch den sowohl die Peptid-Liganden als auch die Tumorspezifität aller Varianten verifiziert wurden.

Durch diese Arbeit wurde bei der Mehrzahl der Patienten unabhängig von der Tumorentität eine breite Vielfalt an nicht-kanonischen HLA-präsentierten Peptiden nachgewiesen, von denen 32 Neoantigen-Kandidaten validiert wurden. Acht dieser 32 validierten Neoantigen-Kandidaten erwiesen sich in Stimulationsexperimenten mit autologen PBMCs und TILs sowie mit allogenen, gesunden Spender-PBMCs als immunogen. Die Mehrheit der gesamten und immunogenen Neoantigene stammten von Varianten aus dem RNA-Mutationsdatensatz, was die Bedeutung von RNA als noch wenig erforschte Quelle von Krebsantigenen verdeutlicht. Darüber hinaus korrelierte die Menge dieser hauptsächlich RNA-basierten immunogenen Neoantigene positiv mit der Gesamtzahl der CD3$^+$ tumorinfiltrierenden T-Zellen.

Diese Arbeit unterstreicht somit die Bedeutung der RNA-basierten Variantenerkennung für die Identifizierung potenziell relevanter Neoantigen-Kandidaten und gibt Vorschläge zur Verbesserung zukünftiger Neoantigen-Identifizierungspipelines und der Neoantigen-Priorisierung für eine potenzielle klinische Anwendung.

1. Introduction

1.1 Multi-omics era of cancer research – a systemic approach

"Those researching the cancer problem will be practicing a dramatically different type of science than we have experienced over the past 25 years. Surely much of this change will be apparent at the technical level. But ultimately, the more fundamental change will be conceptual."

– Hanahan & Weinberg (2000) –

This prognosis established by Hanahan and Weinberg in their famous essay "The Hallmarks of Cancer" over 20 years ago could not be more accurate. As postulated, not only the techniques but also the general understanding and concepts of cancer research have changed tremendously since then and are, of course, still evolving.

The hallmarks, described and later expanded by Hanahan and Weinberg in 2011, highlight the complexity of tumorigenesis with multiple different pathways, processes and mechanisms influencing multiple cellular but also systemic layers. Understanding all these processes requires special techniques to generate large amounts of data and ultimately integrate those, not only within cancer cells but also the surrounding tumor microenvironment (TME). Here, omics technologies provide a great tool that has evolved more and more during the last years, as postulated by Hanahan and Weinberg. Furthermore, the possibility to generate such comprehensive and large data and by that get a more comprehensive knowledge about many processes not only changed the concept of cancer research but also more importantly treatment strategies for patients. In the following I will elaborate about these technical and conceptual changes in more detail.

1.1.1 Technique: Omics techniques and the power of integration

The suffix -omics is used to describe high-throughput technologies and assays that aim at collectively analysing the characteristics, quantities, and functions of pools of biological molecules at different levels. The aim is to provide a holistic/complete data set of a given biological function. Such methods are for example used to not only analyse specific genetic alterations on a small scale (= genetics) but to decipher the complete genome as a whole (= genomics). Several single-omics analysis techniques have been developed to understand different biological functions such as the genome, transcriptome, epigenome, proteome, metabolome, microbiome and lipidome. Some of the single-omics technologies and their applications are listed in Table 1.

Table 1: Different omics techniques and their applications.
HLA, human leukocyte antigen; HPLC, high performance liquid chromatography; IP, immunoprecipitation; MRI, magnetic resonance imaging; MS, mass spectrometry; NGS, next generation sequencing.

Omics	Type	Principle	Application
Genomics	Whole genome sequencing	NGS	Mutation analysis across genome
	Whole exome sequencing	NGS	Mutation analysis across exome
	Targeted sequencing	NGS	Mutation analysis of specific genes/regions
Epigenomics	Methylomics	NGS of bisulfate-treated DNA	DNA methylation pattern across genome
	ChIP-Sequencing	Chromatin-IP & NGS	Epigenetic marks across genome
Transcriptomics	RNA sequencing	NGS	Differential gene expression across genome
	Microarray	Hybridization	Differential gene expression of specific genes
Proteomics	Deep-proteomics	MS	Differential protein abundance across genome
	Reverse-phase protein array	Antibody-based microarray	Differential protein abundance of specific genes
Immunopeptidomics	HLA-presented peptidomics	HLA-IP and MS	Identification and quantification of all HLA-presented protein-peptides across genome
Secretomics	Deep-secretomics	MS	Identification and quantification of all secreted proteins
Metabolomics	Deep-metabolomics	MS	Differential metabolite abundance
Lipidomics	Deep-lipidomics	MS	Identification and quantification of all lipid species
Microbiomics	Deep-microbiotics	NGS	Identification and characterisation of all microorganisms of a given community
Immunomics	System immunology	Several omics techniques	Holistic understanding of the immune system, its functions and regulation
Glycomics	Deep-glycomics	MS	Identification and quantifcation of all sugar species
	Glyco arrays	Lectin + antibody arrays	Identification and quantifcation of all sugar species
	Deep-glycomics	HPLC	Identification and quantifcation of all sugar species
Connectomics	Neural imaging	MRI	Comprehensive maps of connections within an organism's nervous system on a macroscopic scale
	Neural imaging	Electron microscopy	Comprehensive maps of connections within an organism's nervous system on a microscopic scale

Although all these single-omics analysis provide an understanding of their respective biological function, the complex interactions and thus complex phenotypes and molecular changes involved in carcinogenesis such as uncontrolled and sustained proliferation, resisting cell death, angiogenesis, metastasis and immune evasion (Hanahan & Weinberg, 2011) cannot be easily reflected. Reaching these hallmarks involves a series of aberrations in the cellular machinery of cancer cells and whole tissues induced by molecular alterations in the genome, transcriptome, epigenome, proteome, and metabolome. Therefore, integration of several omics data sets of single layers into a multi-omics analysis provides a methodology to gain insights into causal relations of these processes and to understand the underlying biology of such a complex disease as cancer (Menyhárt & Győrffy, 2021). Moreover, such multi-omics analysis can provide a better understanding of prognostic and predictive phenotypes, classify distinct cancer subtypes and can help dissect cellular responses to therapy (Chakraborty et al., 2018). Together with the reduction of costs and processing time for omics analysis, the use of multi-omics for the understanding of these aspects has increased tremendously over the past years.

Therefore, several different mathematical methods have been developed to integrate multi-omics data that are mostly categorized as Bayesian, similarity-based, network-based, fusion-based, and correlation-based and led to the development of many tools and computational frameworks such as iCluster, SALMON, PARADIGM, NEMO and many more. The methodological details and their use in cancer research are extensively described in several reviews (Heo et al., 2021; Menyhárt & Győrffy, 2021; Nicora et al., 2020; Raufaste-Cazavieille et al., 2022; Subramanian et al., 2020) and will not be discussed in more detail here.

Besides the development of mathematical models, several publicly available data bases have been established in the past years that combine a plethora of omics data sets (Raufaste-Cazavieille et al., 2022). The biggest publicly available data base of multi-omics data so far is The Cancer Genome Atlas (TCGA) (Weinstein et al., 2013), that aims at cataloguing alterations in the DNA and chromatin of the cancer-genomes (single nucleotide variation (SNV), copy number variations (CNV), methylation) and linking these aberrations to transcriptomics, proteomics and clinical data. Data on such altered and matched normal specimens are available for over 20,000 subjects and 30 different human cancer types and integrative analysis on these data sets has been performed by multiple groups for different scientific and clinical questions (Akbani et al., 2015; Liu et al., 2018; Mertins et al., 2016; The Cancer Genome Atlas Research Network, 2013).

Mainly, the focus in the past years was set on genomic and transcriptomic data, however also several studies combining genomics and peptidomics, called proteogenomics, have been performed for cancer stratification, treatment monitoring and biomarker analysis (Chen et al., 2020; Krug et al., 2020; Lehmann et al., 2021), besides many others.

However, despite the great impact of multi-omics studies, they are limited by their data quality, the imbalance between availability of different -omics data types in data bases such as TCGA and CPTAC, over- and underrepresentation of certain tumor types in these data bases, the availability and systematization of matching clinical data and the choice of the analysis method or tool (Menyhárt & Győrffy, 2021; Subramanian et al., 2020). For the latter, a detailed benchmarking and generalisation of methods and tools for these analyses could help identify the best tool for each scientific question (Cantini et al., 2021; Tini et al., 2018) and would lead to a better interpretability of data and comparability between studies. Together with the improvements on data visualisation methods and tools (Cerami et al., 2012), a more uniform framework would need to be established that might ease analysis, visualization and interpretation of multi-omics data in a comprehensive/all-inclusive manner to make it easily usable for researchers and ultimately clinicians. This would accelerate the implication of multi-omics data analysis into the daily medical routine tremendously.

Also, methods to integrate the evolving field of single-cell multi-omics or spatial omics technologies (Akhoundova & Rubin, 2022) will provide future challenges but also improvements.

1.1.2 Concept: From single entity to systemic pan cancer approaches

The above-described technical advancements in cancer research using omics data, especially genomics, as well as multi-omics analysis led to novel insight into many different cancer entities. These analyses revealed a large molecular diversity within a same tumor type (Krug et al., 2020; Lehmann et al., 2021; Lindskrog et al., 2021) as well as spatial and temporal heterogeneity of tumors (Dagogo-Jack & Shaw, 2018). Also, several molecular characteristics and common driver events have been identified between cancers of different entity (Dong, 2021; Hoadley et al., 2014). All of this paved the way towards a new concept in understanding and especially treating cancer - as Hanahan and Weinberg postulated back in 2000 - from tumor site classification to pan-cancer molecular classification.

Previously, patients have been treated according to their tumor's entity, mainly defined by phenotypic characteristics of the tumor such as location and cell morphology/histology. With the knowledge from multi-omics studies, patients are nowadays more and more classified and treated according to the molecular characteristics of their tumor rather than its entity (Hoadley et al., 2014). This on the one hand make systemic pan-cancer treatments, that focus on molecular characteristics commonly underlying a range of cancer types, very attractive but on the other hand also offers new therapeutic opportunities for more personalized precision medicine (see Figure 1).

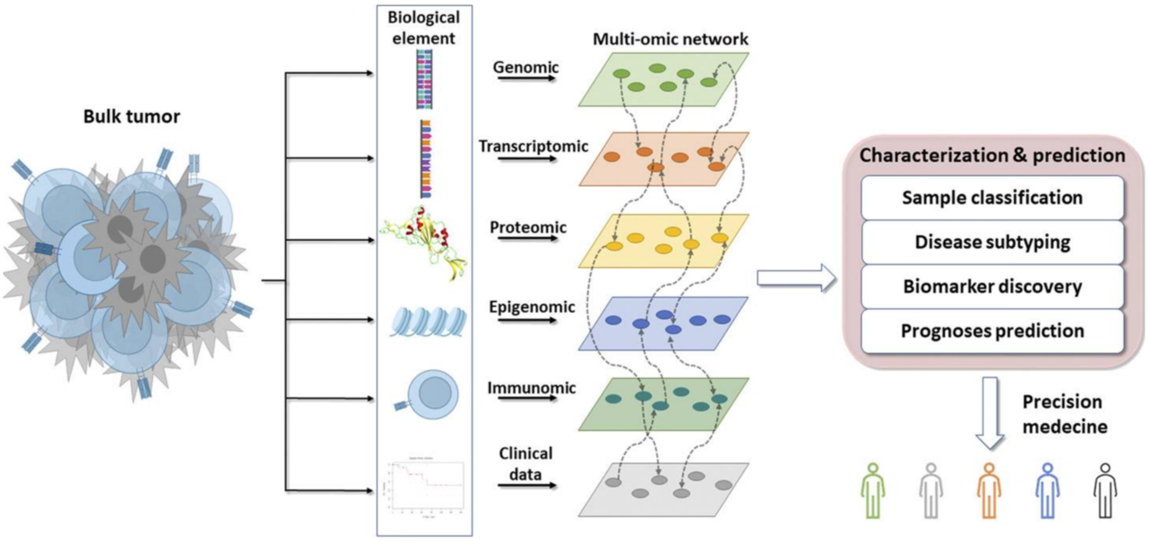

Figure 1: Multi-omics data integration for precision oncology.
The bulk tumor, composed of different cell types and biological elements, is used for a multitude of different omics analysis techniques. Each of these omics analysis gives a unique data set with specific information about the tumor that are combined within a multi-omics network. Novel interactions between interconnected elements can now be identified and used to molecularly characterize and subtype the tumor and ultimately predict prognosis and therapy outcome for each individual patient. (Illustration from Raufaste-Cazavieille, Santiago and Droit, 2022)

Therefore, especially genomic profiling of tumors using genomics and transcriptomics was used by several groups to stratify and treat cancer patients with different types of tumors, especially advanced and rare tumors, according to their mutational profile (Horak et al., 2021; Massard et al., 2017; Pleasance et al., 2022; Rodon et al., 2019; Zehir et al., 2017). These studies and clinical trials already integrated the use of omics data into the clinic, thus influencing direct patient care, and could demonstrate an overall physician-assessed clinical benefit of 46% (Pleasance et al., 2022) and significantly improved disease control rate of 55% (Horak et al., 2021) using genomics-informed precision medicine.

Besides the identification of targetable molecular alterations for treatment of patients, also the discovery of biomarkers associated with prognosis and treatment sensitivity as well as resistance is of great importance. This enables a classification of patients into specific risk-groups and grants the opportunities to provide therapies tailored to the biological characteristics of a specific tumor.

Many different pan-cancer biomarker have been proposed over the past years (Dong, 2021), illustrated by over 5,000 research articles that can be found in PubMed when searching for “pan-cancer biomarker”. However, only few pan-cancer biomarkers have been approved so far by the U.S. Food and Drug Administration (FDA): microsatellite instability-high (MSI-H) and mis-match-repair-deficient (dMMR) became the first examples for a tissue-agnostic biomarker-based approval of the immune checkpoint inhibitor (ICI) pembrolizumab in 2017 (Boyiadzis et al., 2018; Lemery et al., 2017; U.S. Food and Drug Administration, 2017). This was followed by the approval of larotrectinib for patients with neurotrophic receptor tyrosine kinase (NTRK) gene fusion in 2018, where the efficacy was demonstrated in 12 different cancer indications (U.S. Food and Drug Administration, 2018). Most recently also high tumor mutational burden (TMB) was approved as a tissue-agnostic biomarker for the use of pembrolizumab (Marcus et al., 2021).

Taken together, with the help of multi-omics data and pan cancer analysis, a systemic approach to cancer treatment has become increasingly important and will change patient care in the clinics in the future years. Precision oncology will enable a much more personalized entity-independent cancer treatment, where targeted therapies but also most importantly immunotherapies will play a major role.

As a systemic therapeutic approach, where not only the tumor cells themselves but rather the tumor microenvironment and the whole host immune system represent the target of therapy, immunotherapy has likewise become highly important for the treatment of cancer. This represents another paradigm shift in cancer treatment, again emphasizing a more systemic and tissue-agnostic oriented therapy approach.

In the past years, the interaction of the cancer cell and the host immune system were extensively explored, also using multi-omics analyses, to better understand the efficacy, side effects and resistance of such therapies.

In the following section I will go into more detail on immunotherapies and their implication for pan-cancer treatment.

1.2 Immunotherapy in the pan cancer world

Immunotherapy is a field of medicine dedicated to enhancing the inherent capabilities of the immune system in combating cancer, by activating or augmenting mechanisms that are impeded during disease progression. Several different types of immunotherapies have been developed such as immune checkpoint inhibition (ICI)(Topalian et al., 2019), vaccinations using RNA, DNA, peptide and dendritic cell (DC)-based (Palmer et al., 2009), diverse forms of antibody therapies (London & Gallo, 2020; Torres-Jiménez et al., 2022) and adoptive T cell therapy, e.g. using tumor-infiltrating lymphocytes (TILS)(S. A. Rosenberg et al., 1994), chimeric antigen receptor (CAR)-T cells (Schuster et al., 2017) or neoantigen-specific T cells (Morgan et al., 2006).

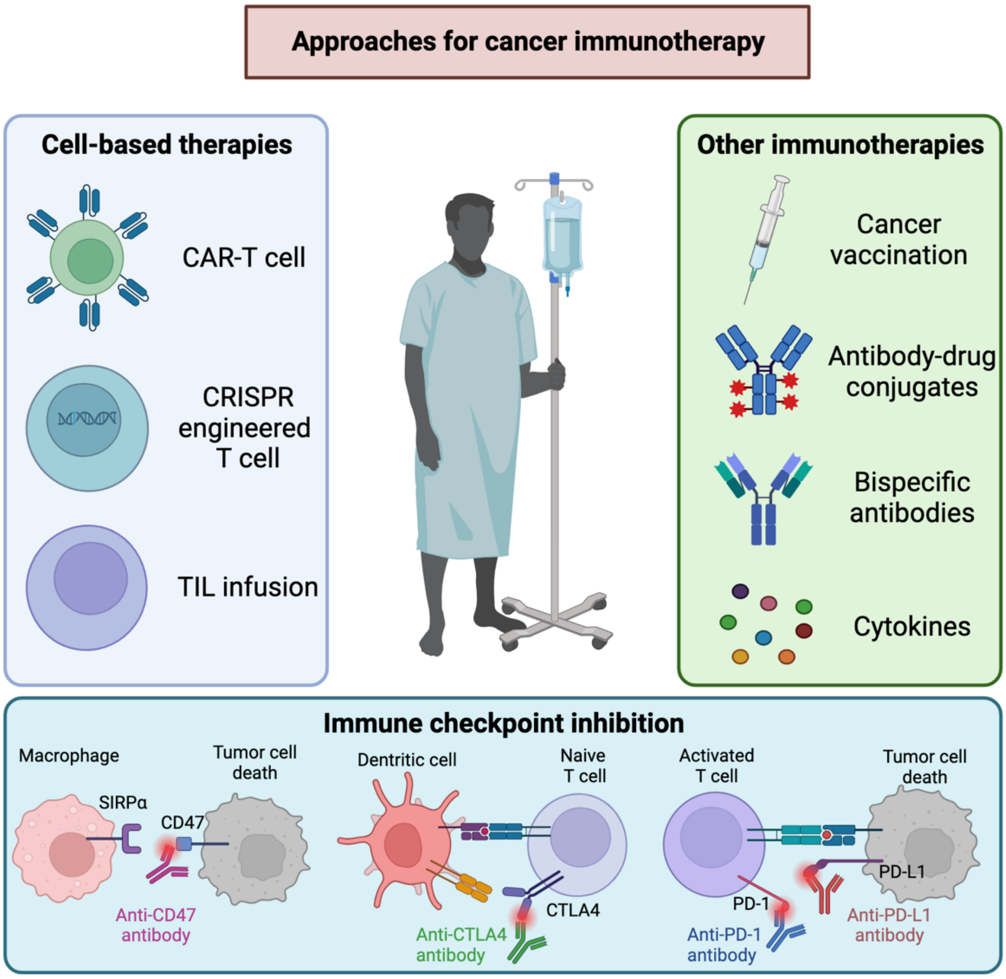

Figure 2: Immunotherapy approaches.
Several different immunotherapy approaches for the treatment of cancer have been developed. Cell-based therapies can be chimeric antigen receptor (CAR)-T cells (Schuster et al., 2017), that are engineered to recognize and kill cancer cells based on specific surface antigens, tumor infiltration lymphocytes (TILs)(S. A. Rosenberg et al., 1994), that are ex-vivo expanded from the patient's tumor and re-infused into the patient, or other engineered T cells carrying e.g. specific T cell receptors (TCRs)(Morgan et al., 2006). Another approach are immune checkpoint therapies that are antibodies which interfere with or block specific immune-related checkpoint molecules on immune cells and cancer cells. By that, immunosuppressive interactions between T cells and other immune cells with the tumor can be blocked (e.g. anti-CD47 antibodies, anti-PD-1 antibodies and anti-PD-L1 antibodies)(Topalian et al., 2015). Furthermore, many other therapies have been developed such as oncolytic viruses (C. Tang et al., 2022), which infect cancer cells specifically and cause lysis of these cells, or antibody-drug conjugates, that specifically deliver cytotoxic drugs to cancer cells based on cancer-specific surface markers such as HER2, CD33 and TROP2 (Shastry et al., 2023). Also, bispecific antibodies can be used, which bind e.g. tumor cells and T cells and by that enable a potential anti-tumor interaction (Ordóñez-Reyes et al., 2022). Finally, also the administration of cytokines for the stimulation of the patient's immune system has been used as immunotherapy such as interleukin-2 and interferon-alpha (Berraondo et al., 2019). (Illustration adapted from Gupta, Mehta and Wajapeyee, 2022, created with BioRender)

The most frequently used immunotherapy type is the previously mentioned ICI, where so called immune checkpoint molecules are targeted to modulate the patients' own immune system to identify and ultimately eliminate cancer cells. The significance of these ICI therapies gained early recognition when the journal Science declared cancer immunotherapy as the "breakthrough of the year 2013" (Couzin-Frankel, 2013). This acknowledgment was further underscored by the Nobel Prize in Physiology or Medicine awarded in 2018 for the discovery of two pivotal immune checkpoint molecules: cytotoxic T-lymphocyte-associated protein (CTLA-4) and programmed cell death protein 1/programmed cell death protein ligand 1 (PD-1 / PD-L1) (Freeman et al., 2000; Ishida et al., 1992; Leach et al., 1996).

The first of such therapeutic antibodies was Ipilimumab (anti-CTLA-4 monoclonal antibody) that was FDA-approved in 2011 for the treatment of malignant melanoma (Cameron et al., 2011) followed by Pembrolizumab (anti-PD-1 monoclonal antibody) in 2014 (Robert et al., 2014). Since then, the approval of Pembrolizumab has been expanded to further cancer types and several novel ICI agents have been developed and approved for anti-cancer treatment (Davis & Patel, 2019; Vaddepally et al., 2020) (summarized in Table 2).

Table 2: FDR approved immune checkpoint inhibitors in order of first approval.
All monoclonal antibodies used as immune checkpoint inhibitors in order of approval and the respected approved cancer types also in order of approval (as of December 2023). Notes: [1] in combination with nivolumab, [2] PD-L1 as biomarker, [3] also in combination with ipilimumab. cSCC, cutaneous squamous cell carcinoma; CTLA-4, cytotoxic T-lymphocyte-associated protein 4; dMMR, mismatch repair deficient; HCC, hepatocellular carcinoma; HNSCC, head and neck squamous cell cancer; mAb, monoclonal antibody; mCRC, metastatic colorectal cancer; MSI-H, microsatellite instability-high; NSCLC, non-small-cell lung cancer; PD-(L)1, programmed death (ligand) 1; PMBCL, primary mediastinal large B-cell lymphoma; RCC, renal cell carcinoma; SCLC, small-cell lung cancer, TMB-H, tumor mutational burden high. (Massard et al., 2016; Migden et al., 2018; U.S. Food and Drug Administration, 2023; Vaddepally et al., 2020).

Target	Name	Type	First approval date	First approval study	Approved cancer types
CTLA-4	Ipilimumab	mAb	28 March 2011	MDX010-020	HCC[1], mCRC (dMMR, MSI-H)[1], Melanoma[1], mesothelioma[1], NSCLC[1,2], RCC[1]
PD-1	Pembrolizumab	mAb	04 Sep 2014	NCT01295827	Advanced gastric carcinoma[2], biliary tract cancer, cervical cancer[2], cSCC, HCC, HNSCC[2], Hodgkin's lymphoma, melanoma, Merkel cell carcinoma, NSCLC[2], PMBCL, RCC, solid tumors with MSI-H or dMMR or TMB-H, triple-negative breast cancer[2], urothelial cancer[2]
	Nivolumab	mAb	22 Dec 2014	CheckMate-037	Gastric cancer, HCC[3], HNSCC, Hodgkin's lymphoma, mCRC (MSI-H or dMMR)[3], melanoma[3], mesothelioma1, NSCLC[2,3], RCC[3], SCLC, urothelial cancer
	Cemiplimab	mAb	28 Sep 2018	NCT02383212, NCT02760498	Basal cell carcinoma, cSCC, NSCLC[2]
PD-L1	Avelumab	mAb	18 Nov 2015	JAVELIN Merkel 200	Merkel cell carcinoma, RCC, urothelial carcinoma
	Atezolizumab	mAb	18 May 2016	IMvigor210	HCC, melanoma, (N)SCLC[2], triple-negative breast cancer[2], urothelial carcinoma[2]
	Durvalumab	mAb	01 May 2017	MEDI4736	HCC, (N)SCLC, urothelial carcinoma[2]

The efficacy of ICI in the treatment of an increasing number of cancer entities is well established (Eggermont et al., 2016; Topalian et al., 2019) and highlights their great potential as entity-independent systemic therapies. However, side effects such as immune-related adverse events are often reported (Postow et al., 2018) and many patients show primary or acquired resistance to ICI (Restifo et al., 2016).

Therefore, several biomarkers have been proposed to predict therapy response and survival, preferably irrespective of tumor entity. These include TMB (Klempner et al., 2020; Rizvi et al., 2015; Snyder et al., 2014), PD-L1 expression (Patel & Kurzrock, 2015; Topalian et al., 2012), immune cell infiltration (Galon et al., 2006; Leffers et al., 2008; Oshi et al., 2020) and other tumor- and host-related factors (Havel et al., 2019). However, not all of these biomarkers show a pan-cancer implication, exemplarily highlighted by the expression status of PD-L1 that has been demonstrated to have limitations as predictive biomarker, especially seen in non-small-cell lung cancer (NSCLC) (Carbone et al., 2017; Davis & Patel, 2019; Reck et al., 2016). In contrast, a more robust pan-cancer biomarker is the TMB which is playing an important role for the response to ICI in cross disease cancer entities as shown by several studies (Litchfield et al., 2021; Samstein et al., 2019) also using multi-omics data sets (Pender et al., 2021). Moreover, MSI-H or dMMR is associated with high mutational load and predictive for response to immunotherapy (Le et al., 2015, 2017). As previously described, all of these three markers became the first examples for a tissue-agnostic biomarker-based approval of ICI by the FDA (Boyiadzis et al., 2018; Marcus et al., 2021).

In addition to these tumor intrinsic characteristics, also the immune microenvironment is of central importance to understand patients' survival and response to immunotherapy, especially in combination with the above-mentioned factors (Hiam-Galvez et al., 2021). The immune composition of a tumor is extremely heterogeneous within and between patients and often also independent of the tumor stage and entity (Van den Eynden et al., 2019). A number of factors have been identified associated with anti-tumor immunity in defined entities such as the degree, composition, abundance and location of immune cell infiltration, the expression of checkpoint molecules and the phenotypic state of tumor-infiltrating lymphocytes (TILs) (Oliveira et al., 2021; Riaz et al., 2017; Rooney et al., 2015; Zaretsky et al., 2016) . However, although pan-cancer studies exist (Pender et al., 2021), cross-entity analyses to identify hallmarks of cancer independent of the tumor origin are still limited (Lowery et al., 2022; Samstein et al., 2019).

Multi-omics pan-cancer studies are therefore of great importance to include all potentially important biological and clinical factors into the detection of potent biomarkers. This may be relevant to develop entity-agnostic treatment approaches based on common targets and an improved understanding of key factors relevant for survival when administering immunotherapies.

Furthermore, several other immune checkpoint molecules are investigated as potential targets such as TIM-3 and LAG-3 (Qin et al., 2019) and other immunotherapy types are extensively explored to provide alternatives to ICI and thus provide therapy options for patients not eligible or non-responsive to ICI.

1.2.1 Beyond checkpoint inhibition: Cellular immunotherapies

Immune checkpoint inhibition re-activates already existing T cells to enable their anti-tumor response. The requirement for the success of such a therapy is that (enough) T cells are present within the patient that actually can recognize the tumor cells as such. This recognition is based on the ability of immune cells to identify tumor-associated antigens (TAAs) and tumor-specific antigens (TSAs) in the form of cytolytic peptides presented via human leukocyte antigen (HLA) molecules on the surface of several different cell types.

1.2.1.1 Cancer antigens

TAAs are self-antigens that are typically found at normal physiological levels in one or even multiple tissues but can be over-expressed in tumor cells, inducing an immune response (Haen et al., 2020). Carcinoembryonic (CEAs), cancer-testis (CTAs) and viral antigens are specific subtypes of TAAs. CEAs are expressed in early embryonic development, while CTAs are primarily expressed in the germ cells of the testes, however both types of antigens have been found to be aberrantly expressed by cancer and can be used as targets for therapy (Meng et al., 2021). Viral antigens are of importance in virally induced tumors such as head and neck cancer often induced by the human papilloma virus (Conarty & Wieland, 2023; Julian et al., 2021). TAAs have the advantage that they can be found within a larger number of patients but on the other hand have the disadvantage that they are also found on healthy tissue and thus targeting them can lead to unwanted side effects.

TSA, in contrast, are exclusive to the tumor cells as they arise during carcinogenesis and represent new and foreign epitopes for the immune system, called neoantigens. Against these neoantigens T cells can show very strong immune responses as they are not subject to the negative thymic selection, like normal self-antigens. T cells develop within the thymus, each cell expression a unique T cell receptor (TCR) of a single specificity generated via VDJ-recombination. During cell development each of these TCRs is then tested for recognition of self-antigens within the thymus and depleted if a reactivity is observed (thymic depletion). With this negative selection process, only those T cells not recognizing self-antigens survive. However, these cells do recognize any other non-self-antigen with high affinity and avidity. Thus, T cell responses towards TSAs are highly specific to the tumor and in principle do not affect normal tissue which makes them very attractive for therapy purposes. (Chaplin, 2010)
However, TSAs are often specific to a single tumor in a single patient making therapies targeting TSAs highly personalized (Schumacher & Schreiber, 2015).

In cases where ICI fails to be effective, functional tumor-specific T cells might not be present within a patient or the tumor due to several possible reason. For example, just by chance no T cells specific for the tumor antigens is present within the patient, or the T cells present within the patient are already terminally exhausted due to e.g. chronic weak stimulation by the tumor (Jiang et al., 2021) and cannot be re-activated by ICI. Therefore, different therapy strategies have been developed to circumvent a physiological lack of tumor-reactive T cells within a patient that will be described in more detail in this section.

The different sources of tumor-antigens/neoantigens and strategies to identify them are described in the sections below (see 1.2.2 and 1.2.3).

1.2.1.2 Peptide- and DC-based vaccination

One approach to prime and activate the patient's own T cells towards their tumor is vaccination. This approach can be used as a preventative or active therapy for the treatment of cancer and many different vaccination strategies have been developed and clinically tested over the past years. They can be classified into vector-based vaccines (e.g. viral, bacterial or yeast) (Pollack, 2018; Somaiah et al., 2019), peptide/protein-based vaccines (Hilf et al., 2019; Kruit et al., 2013; Ott et al., 2017) and cellular-based vaccines (e.g. whole tumor lysates or peptide-loaded DCs) (Engell-Noerregaard et al., 2009; NCT02334735, 2022; Palmer et al., 2009).

For each strategy the choice of antigen(s) to be targeted is of great importance. Often TAAs such as the CTAs MAGE-A, NY-ESO-1, MART-1 and the glioma-associated antigen WT1 and several more have been used as targets (see Table 3), individually and also in combination, however with variable efficacy (Cebon et al., 2014; Meng et al., 2021; Somaiah et al., 2019). Also, more personalized vaccines against TSA such as neoantigens arising from the patients' somatic mutations have been developed, exemplified by a clinical trial showing feasibility, safety, and immunogenicity of a vaccine targeting up to 20 predicted personalized tumour neoantigens in melanoma (Ott et al., 2017).

1.2.1.3 Adoptive T-cell therapy

Another approach to revive the patients' immune system to fight cancer is to use cellular therapy instead of monoclonal antibodies or vaccination. In this type of therapy tumor-reactive T cells are transferred into the patient that can identify tumor-related antigens (TAAs and TSAs) on the cancer cell and thus promote an immune response towards the tumor ultimately leading to cancer eradication. Three different approaches have been used in the past years, TIL transfer, CAR-T cell therapy and TCR-transgenic T cell therapy.

The first approach of TIL transfer goes back to 1994 where Rosenberg and colleagues isolated, *in vitro* expanded and re-infused TILs from melanoma patients together with interleukin (IL) 2 and achieved a tumor regression in 34% of patients (S. A. Rosenberg et al., 1994). However, the immune cell composition and T cell specificities of infused cell products are heterogeneous and largely unknown. Therefore, several additional studies have been performed in various entities where tumor-reactive TILs were enriched to improve anti-tumor reactivity. One method is to enrich for cytotoxic CD8$^+$ T cells (Chandran et al., 2015; Dudley et al., 2013) or for tumor-antigen specific TILs by co-cultures with engineered DCs (Tran et al., 2016; Zacharakis et al., 2018), however mixed results have been observed. Therefore, the identification of potential tumor-specific peptides presented on the cancer cell surface is of great importance and the characteristics and identification methods for these antigens will be discussed in more detail in the subsequent chapters (see section 1.2.2 and 0)

In contrast to TIL infusions, where the exact targets of T cells largely remain unclear, engineered T cells with CARs or transgenic TCRs represent a more specific approach as these artificial receptors are specifically designed to bind a known target/antigen on the tumor cell.

CARs consist of an intracellular signaling domain, mirroring the structure of TCR constant regions, and, crucially, a target-specific extracellular domain. This extracellular domain is primarily derived from the variable region of an antibody with the desired target-specificity (reviewed extensively in (June et al., 2018; Sadelain et al., 2013; R. C. Sterner & Sterner, 2021). Cloned into the patients' own peripheral blood mononuclear cells (PBMCs) *ex vivo*, these T cell products can effectively destroy cancer cells when re-infused into the patient. Due to the antibody-like binding properties of these CARs, CAR T cells can bind independent from the HLA receptor to their antigens and thus mainly stably expressed surface proteins are targeted rather than HLA-presented peptides. Therefore, CAR T cells do not require matching to a patient's haplotype, reducing the need for highly personalized patient selection. Additionally, this flexibility allows them to target tumor cells with down-regulated HLA expression and/or impaired proteasomal antigen processing. One prominent example of such a CAR-therapy is targeting the surface molecule CD19 mainly expressed on B cells and thus is highly successfully used for the treatment of B-cell malignancies, approved by the FDA in 2017 (Maude et al., 2018; Neelapu et al., 2017; Schuster et al., 2017). Numerous additional CARs, designed to target a range of surface antigens, have been developed (Carpenito et al., 2009; Hudecek et al., 2013; Louis et al., 2011). However, side effects are frequently observed, such as cytokine storms and on-target off-tumor responses (Kochenderfer et al., 2012; Maude et al., 2018; J. H. Park et al., 2018; R. M. Sterner & Kenderian, 2020), often attributed to the non-physiological nature of CARs. Furthermore, identifying the right surface marker that is specific for cancer cells, meaning a) ideally not or only rarely expressed on normal/non-tumor cells and b) ubiquitously expressed on all cancer cells of a tumor and even on different cancer types, hold a huge challenge for the successful development and clinical implementation of CAR therapies.

A more physiological engineered T cell therapy approach is the use of transgenic TCRs of defined specificity that are cloned into the patients T cells (Cole et al., 1995; Dembić et al., 1986). TCRs in contrast to CARs recognize antigens, in the shape of peptides, that are bound and presented by the patients HLA molecules, making the therapy restricted to specific HLA alleles and potentially even restricted to a single patient. However, in this case, the recognized antigens are peptides derived from cytoplasmatic proteins, significantly broadening the spectrum of potential antigens for TCR-based therapy, and thus presenting a notable advantage when compared to CAR therapy. The targets for such TCR-based therapies can be TAAs and TSAs, either one having their specific characteristics, as described before. The first clinical application were transgenic T cells specific for the CTA MART-1 that showed encouraging anti-tumor efficacy (Morgan et al., 2006). Several further clinical trials against a variety of TAAs and also TSAs have been completed or are currently still ongoing in liquid and solid tumor as summarized in Table 3. However, also for this type of immunotherapy severe side effects due to off-target activity of the TCRs have been observed (Johnson et al., 2009; Morgan et al., 2013; Parkhurst et al., 2011). These might be in part explained by the expression of TAAs on normal tissue or by newly developed autoimmune reactivity of the T cells due to the exogeneous TCR chains forming unwanted pairs with the natural/endogenous TCR chains (Ferrara et al., 2010). Therefore, current research is focusing on improving the safety of such therapies by e.g. enhancing the exogenous TCR dimer formation (Sebestyén et al., 2008) or by depleting the endogenous TCR (Okamoto et al., 2009).

Table 3: Overview about recently completed and ongoing clinical trials using TCR-transgenic T cells.
ALL, acute lymphocytic leukemia; AML, acute myeloid leukemia; CML, chronic myeloid leukemia; CMV, cytomegalovirus; DLBCL, diffuse large B cell lymphoma; HNSCC, head and neck squamous-cell carcinoma; HPV, human papilloma virus; NSCLC, non-small-cell lung cancer. (*ClinicalTrials.Gov*, 2023).

Target	NTC number	Biological agent	Disease	Phase	Status
CD19	NCT04323657	autologous CD19-specific TCR-T cell (TC-110)	non-Hodgkin lymphoma, ALL, DLBCL, primary mediastinal large B cell, mantle cell and follicular lymphoma	I/II	active
CEA	NCT00923806	autologous anti-CEA TCR-engineered PBMCs PG13-CEA_TCR	metastatic cancer (HLA-A*0201 positive)	I	unknown
CMV	NCT02988258	allogenic CMV-TCR transduced donor-derived T cells	haematological malignancies, CMV infection	I	suspended
EBV	NCT03648697	anti- EBV-TCR-T (YT-E001) cells (LMP1, LMP2 and EBNA1)	nasopharyngeal carcinoma	II	unknown
	NCT04139057	autologous EBV-specific T cell	HNSCC	I/II	unknown
	NCT04509726	autologous LMBP2-specific T cell	nasopharyngeal carcinoma	I/II	acitve
gp100	NCT00923195	anti-gp100:154 TCR-engineered PBMC	skin cancer, melanoma	II	completed
HA-1	NCT03326921	allogenic CD8$^+$ and CD4$^+$ donor memory T-cells-expressing HA1-specific TCR	leukemia	I	suspended
HBV	NCT02686372	autologous anti-HBV TCR-T cells	hepatocellular carcinoma	I	completed
	NCT02719782	autologous anti-HBV TCR-T cells	recurrent hepatocellular carcinoma	I	unknown
	NCT05339321	autologous anti-HBV TCR-T cells (SCG101)	hepatocellular carcinoma	I	active
HPV-E6	NCT02280811	autologous HPV-16 E6-specific T cells	vaginal, cervical, anal, penile and oropharyngeal cancer	I/II	completed
	NCT03578406	autologous HPV-16 E6-specific T cells	cervical cancer, HNSCC	I	unknown
	NCT05357027	autologous HPV-16 E6-specific T cells (TC-E202)	cervical carcinoma	I/II	active
HPV-E7	NCT02858310	autologous E7-specific T cells	cervical intraepithelial neoplasia, vulvar neoplasms, vulvar diseases	I/II	active
	NCT05639972	autologous E7-specific TCR-T cells	HPV-related malignancies	I/II	active

Target	NTC number	Biological agent	Disease	Phase	Status
KRAS	NCT03190941	autologous anti-KRAS G12V mTCR PBL	pancreatic, gastric, gastrointestinal, colon and rectal cancer	I/II	active
	NCT03745326	autologous anti-KRAS G12D mTCR PBL	pancreatic, gastric, gastrointestinal, colon and rectal cancer	I/II	active
LAGE-1a	NCT04526509	autologous LAGE-1a-specific CD8-positive T Lymphocytes	locally advanced or unresectable malignant neoplasm, sarcoma	I	active
MAGE	NCT02111850	Anti-MAGE-A3-DP4 TCR-engineered PBMC	cervical, renal, urothelial and breast cancer, melanoma	I/II	completed
	NCT03247309	autologous MAGEA4/8 TCR-engineered PBMCs (IMA201-101)	solid cancer (expression of MAGEA4 and/or -8)	I	active
	NCT03441100	autologous MAGEa1 TCR-engineered PBMCs (IMA202)	solid cancer (expression of MAGEA1)	I	active
	NCT04729543	autologous MAGE-C2/HLA-A2 TCR T cells (MC2 TCR T cells)	advanced melanoma, HNSCC	I/II	active
MART-1	NCT00091104	anti-MART-1 TCR-engineered lymphocytes	melanoma	I	completed
	NCT00509288	autologous anti-MART-1 F5 TCR-engineered TILs	skin cancer, melanoma (HLA-A*0201 positive)	II	completed
	NCT00923195	Anti-MART-1 F5 TCR-engineered PBMC	skin cancer, melanoma	II	completed
Mesothelin	NCT03907852	autologous anti-mesothelin engineered T cells (TC-210)	advanced mesothelin-expressing cancer	I/II	active
	NCT05451849	autologous anti-mesothelin engineered T cells (TC-510)	advanced mesothelin-expressing cancer	I/II	active
NY-ESO	NCT01567891	autologous NYESO-1c259 engineered T cells	ovarian cancer	I/II	completed
	NCT01967823	Anti-NY ESO-1 mTCR PBL	melanoma, meningioma, NSCLC, breast and hepatocellular cancer	II	completed
	NCT02650986	autologous NY-ESO-1 TCR/dnTGFbetaRII transgenic T cells	advanced and/or metastatic solid cancer	I/II	active
	NCT02775292	autologous NY-ESO-1-specific engineered PBL	metastatic cancer, adult and childhood solid cancer	I	completed
	NCT03029273	autologous anti-NY-ESO-1 TCR-transduced T cells	NSCLC	I	unknown
	NCT03240861	autologous NY-ESO-1 TCR-engineered PBMCs and stem cells	locally advanced or unresectable malignant neoplasm, sarcoma	I	active
	NCT03462316	autologous anti-NY-ESO-1 TCR -transduced T cells	bone and soft tissue sarcoma	I	active
	NCT03691376	autologous NY-ESO-1-specific CD8-positive T Lymphocytes	platinum-resistant/-sensitive, recurrent and refractory fallopian tube, ovarian and primary peritoneal carcinoma	I	active
	NCT04526509	autologous NY-ESO-1-specific CD8-positive T lymphocytes	locally advanced or unresectable malignant neoplasm	I	active
	NCT05296564	autologous NY-ESO-1-specific engineered lymphocytes (HBI 0201-ESO TCRT)	metastatic sarcoma, melanoma, breast cancer, NSCLC, ovary cancer, bladder urothelial carcinoma, neuroblastoma	I/II	active
	NCT05648994	autologous NY-ESO-1-specific TCR-engineered PBL	solid tumor	I	not yet recruiting
P53	NCT00393029	anti-p53 TCR transduced PBMC	metastatic cancer, overexpress p53	II	
PRAME	NCT02743611	autologous PRAME-specific TCR-engineered T cells (BPX-701)	AML, uveal melanoma	I/II	unknown
	NCT03503968	autologous PRAME-specific TCR-engineered T cells (MDG1011)	myeloid and lymphoid neoplasms	I/II	active
	NCT03686124	ACTengine® IMA203 and IMA203CD8$^+$ products	solid cancer (expression of PRAME)	I	active
SL9	NCT00991224	WT-gag-TCR or α/6-gag-TCR modified T cells	HIV infection	I	completed
WT-1	NCT02550535	autologous WT1-TCR gene-transduced T cells	AML	I/II	completed
Multiple TAAs	NCT04284228	donor-derived TAA-specific CD8$^+$ T cells (NEXI-001 T Cells)	AML, CML	I/II	completed
	NCT03652545	Autologous TAA-specific TCR-T cells	Brain tumor	I	active
Personalized TSAs/ neoantigens	NCT03412877	autologous neoantigen specific TCR-T cell	metastatic cancer	II	active
	NCT03778814	autologous tumor specific TCR-T cells	solid tumor, NSCLC	I	active
	NCT03970382	autologous neoantigen specific TCR-T cells	solid tumor	I	suspended
	NCT04520711	autologous tumor specific TCR-T cells	malignant epithelial cancers	I	active
	NCT05124743	autologous neoantigen specific TCR-T cell drug product	ovarian, endometrial, colorectal and pancreas cancer, cholangiocarcinoma, NSCLC	I/II	active
	NCT05194735	autologous neoantigen specific TCR-T cell drug product	gynecologic, colorectal, pancreatic, ovarian, lung and endometrial cancer, NSCLC, cholangiocarcinoma	I/II	active
	NCT05349890	autologous engineered T cells targeting tumor-specific antigens	malignant epithelial cancers	I	active

As highlighted within this section, the identification and selection of targetable TAAs and also TSAs is of utmost importance for the development and efficacy improvement of these therapies. Although many of the mentioned treatments are only investigated within a defined entity, TAAs but also potentially shared TSAs can provide interesting pan-cancer targets, which need to be explored further. However, neoantigens can represent promising targets for more personalized immunotherapy approaches as they are tumor specific as shown for defined tumor entities and identification as well as characterization of such neoantigens is becoming increasingly important (Bräunlein et al., 2021; Tran et al., 2015; Verdegaal et al., 2016)

In the next sections I will therefore focus on different sources of neoantigens and strategies for their identification.

1.2.2 Sources of neoantigens

Neoantigens as TSA can arise from various sources and by different mechanisms (see Figure 3). Association of mutational burden with response to ICI highlights that neoantigens arise as a consequence of DNA damage and genetic alterations (Schumacher & Schreiber, 2015). For this reason, the major source of neoantigens were thought to be somatic mutations on coding exons, focusing on SNVs that directly change the sequence of a peptide making it a neoantigens (Bassani-Sternberg et al., 2016). Of course, also other genetic events on DNA level such as nucleotide duplications, insertions and deletions (InDels) can directly and indirectly lead to the formation of neoantigens, the latter via frame shifts (Schwitalle et al., 2008).

More recently, it has been reported that neoantigens can arise not only from somatic mutations in coding exons but also from variants found in non-coding transcripts, including pseudogenes, long non-coding RNAs (lncRNAs), and regions with unknown functions (Chong et al., 2020; Laumont et al., 2018). This broadens the potential to identify targetable neoantigens, as 99% of cancer mutations can be found in non-coding regions (Khurana et al., 2016). Moreover, as much as 75% of the total genome can undergo transcription and potential translation, while the entire exome constitutes only 2% (Djebali et al., 2012). As an example, pseudogenes can recover their lost protein-coding function in cancer (Poliseno et al., 2015) and lncRNAs were found to be translated, however not generating functional proteins but potentially immunogenic peptides that were found to be presented on the tumor surface (Chong et al., 2020). Furthermore, variants in genes coding splicing factors or variants in splice sites can lead to intron inclusions and by that neoantigens from intronic regions compose another source of potentially immunogenic tumor-specific peptides (Bigot et al., 2021; Smart et al., 2018).

In addition to variants on DNA level, recent research also demonstrated the importance of neoantigens arising due to variants on RNA level. However, these peptides are often referred to as aberrantly expressed or non-canonical peptides rather than neoantigens as they are not mutated directly and can be tumor-specific but also tumor-associated. Examples of RNA based variants leading to aberrant peptides are gene fusions, dysregulation of transposable element expression, alternative splicing variants and post-translational modifications (Cheng et al., 2022; Laumont et al., 2018; J. Park & Chung, 2019; Rathe et al., 2019). Furthermore, recent attention has been directed towards a more in-depth investigation of RNA processing events, such as RNA editing, as a source for aberrantly expressed peptides and neoantigens (Zhang et al., 2018).

RNA editing is a prevalent post-transcriptional mechanism that induces specific and consistent nucleotide changes in selected RNA transcripts in normal cells (Bazak et al., 2014) but this mechanism is also implicated in disease pathogenesis and undergoes alterations in the context of cancer (Han et al., 2015; Peng et al., 2018; Roth et al., 2018). The most common RNA editing type is the conversion of adenosine (A) to inosine (I) (A-to-I editing, seen as A to guanosine (G) (A-to-G) on RNA sequencing data), which is catalyzed by the enzymes of the Adenosine Deaminases Acting on RNA (ADARs) family. A-to-I editing can affect coding RNAs but also non-coding RNAs such as microRNA (miRNA) (Nishikura, 2015) leading to alterations within their sequence and by that causing the dysregulation of several cellular processes in cancer (H. Wang et al., 2021). Many tools have been developed to identify these editing events and about 15.6 million editing sites have been reported in the biggest RNA editing data base covering almost all identified editing sites called REDIportal (Mansi et al., 2021), of which most reside in non-coding RNA regions. Furthermore, RNA editing events have been linked to the diversification of the cancer proteome in recent publications (Peng et al., 2018; Yang & Nam, 2020), thus implicating that edited RNAs can in fact be translated. RNA variants resulting from editing events have been subject to more detailed investigation as a potential source of aberrantly expressed peptides (Zhang et al., 2018; Zhou et al., 2020). Indeed, Zhang *et al.* was already successful in identifying reactive RNA editing peptides (A-to-I editing) that elicit an anti-tumor immune response, although only for two peptides from one editing site in the CCNI gene within melanoma TILs (Zhang et al., 2018). The regulation of RNA is controlled by *cis* regulatory elements and *trans* regulatory factors, a process frequently disrupted by somatic mutations or influenced by oncogenic signaling (Obeng et al., 2019). Thus, antigens arising from cancer-associated RNA editing may, to a certain extent, constitute authentic neoantigens, making them highly relevant for targeted cancer immunotherapy.

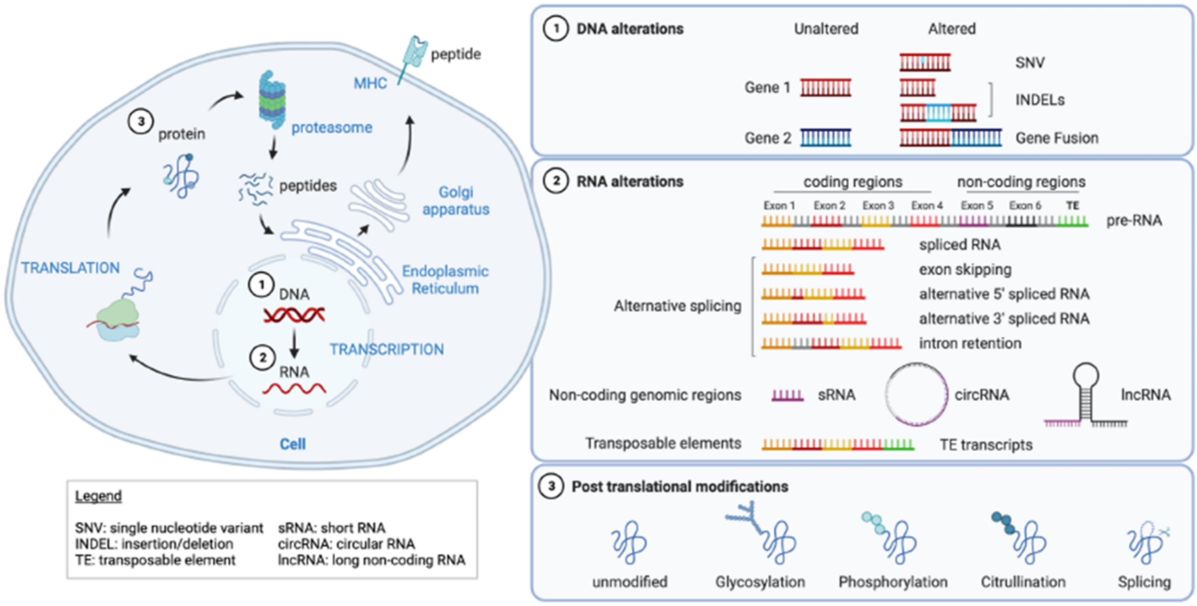

Figure 3: Sources of neoantigen candidates.
Neoantigens can arise from several different sources. (1) DNA alterations can lead to single-nucleotide variations (SNV), insertions or deletions (INDELs) or gene fusions. All these variants lead to the formation of a new DNA sequence, however only those variants also leading to a new amino acid sequence can lead to potential neoantigens. These processes mainly take place in the nucleus and directly chance the genomic information of the cell, thus are kept once the cell divides. (2) Alterations on RNA level can also be responsible for the generation of new amino acid sequences. Potential non-self peptides can be formed by several different mechanisms such as alternative splicing, RNA editing or by the abnormal translation of non-coding regions or transposable elements within the nucleus of a cell. (3) Also post translational modifications such as phosphorylation and glycosylation can lead to aberrant peptides that can be recognized as non-self by the immune system and thus representing neoantigens. These processes mainly happen in the cytosol before proteasomal degradation and loading onto the HMC molecule. (Illustration from Capietto, Hoshyar and Delamarre, 2022)**.**

Irrespective of the source of neoantigen or aberrantly expressed peptide, not only the question of tumor specificity but also clonality of the variant play a crucial role for the decision of a therapeutic use of these peptides. As discussed in the section above, TSAs are highly promising for personalized treatments and may induce stronger T cell responses, however TAAs have successfully been used for therapy as well, as they provide a higher potential for pan-cancer targets. Also, tumor heterogeneity plays an important role, as variants leading to targetable peptides should preferably be clonal and present on as many tumor cells as possible for a good tumor response (McGranahan et al., 2016).
However, detection of these neoantigens/aberrantly expressed peptide in the first place and defining their reactivity in the second place are the most challenging parts.

1.2.3 Identification methods of neoantigens

For the identification of neoantigens and aberrantly expressed peptide (for better readability only referred to as neoantigens in the following) from patient tumor samples two mayor strategies have been used in the past years: reverse immunology that relies on prediction algorithms and proteogenomics that includes mass spectrometry analysis for direct neoantigen identification.

The basis for both strategies is the identification of variants/alterations from genomics and transcriptomics data by next generation sequencing. Therefore, whole genome sequencing (WGS) or whole exome sequencing (WES) of tumor and normal tissue derived DNA and/or RNA sequencing (RNA-seq) of tumor tissue derived RNA is performed. Subsequently, both normal and tumor reads are aligned to the human reference genome and several different variant-calling algorithms can be used to identify DNA mutations and other genetic variants. The identification of RNA variants however is more complex, as here often no matched normal tissue is available or normal tissue might not represent actual wild-type (wt) characteristics as gene transcription is highly dynamic. Furthermore, alignment of reads is made more difficult due to alternative splicing or random errors introduced during transcription or library preparation, leading to a potentially higher false positive rate. Therefore, RNA-seq is often used in combination with genomics analysis (Hashimoto et al., 2021) and helps in detecting variants exceeding a certain expression threshold or those that were missed on DNA level, but also to broaden the landscape of possible variants as described before. All variants are then combined into a patient and even tumor specific variant data base that is used as the basis for either prediction or proteogenomics.

In silico predictions of potential neoantigens is most commonly used where prediction algorithms try to foresee the potential of a given mutated peptide to be presented on the tumor cell surface. This is mainly done via the binding prediction of the peptide towards the patients’ HLA class I and class II molecule where neuronal-network algorithms such as MHCflurry and NetMHC are used (Andreatta & Nielsen, 2016; O’Donnell et al., 2018). Other tools to predict cellular processes such as peptide processing (e.g. SYFPEITHI, NetChop (Keşmir et al., 2002; Rammensee et al., 1999)) or transport (e.g., NetCTL (Larsen et al., 2007)) have been developed, however most of the algorithms cannot reflect the complexity within the cell accurately and can be incomplete or biased due to the training data available. As training data is mainly available for the more frequently expressed major histocompatibility complex (MHC) types (Gonzalez-Galarza et al., 2020a), algorithms often have problems in correctly predicting binding affinities to rare MHC alleles. Another approach to prioritize predicted neoantigen candidates was suggested, that compares the predicted binding affinity of each neoantigen candidate to their canonical counterpart and ranks them according to the improved binding to the HLA molecules induced by the mutation, called differential agretopicity index

(DAI)(Duan et al., 2014). These tools have shown success in identifying several neoantigens, although only a small number of these neoantigens were actually identified as immunogenic (Cohen et al., 2015; McGranahan et al., 2016; Tran et al., 2015). As all prediction approaches are somehow using HLA-peptide binding affinity prediction, the selection of neoantigens is often limited to strong binders with high affinities (often set to max. 500nM) and by that often oversees potential immunogenic peptides with weaker binding affinities (Bräunlein et al., 2021; Gros et al., 2016).

A much more direct identification method is the proteogenomics approach, where HLA-bound peptides on the tumor surface can directly be identified. Therefore, the immunopeptidome of a tumor is analyzed by immunoprecipitation of peptide-HLA-complexes (pHLA) from tumor lysates followed by a high-resolution mass spectrometry (MS) analysis (Bassani-Sternberg et al., 2015). All naturally presented peptides are then searched against the previously defined tumor mutations to identify presented neoantigens (Bassani-Sternberg et al., 2016). Several algorithms for the matching of peptide MS spectra with the potential amino acid sequences have been developed and well established such as MaxQuant and pFind (Chi et al., 2018; Cox & Mann, 2008), however also novel tools such as the artificial intelligence algorithm Prosit (Gessulat et al., 2019; Wilhelm et al., 2021) are developed to deal with the increasing size of the search space and increasing amount of data. Although the clear advantage of the proteogenomics approach is the direct identification of presented peptides, the method is limited by the MS analysis itself and the power an accuracy of the peptide-spectra matching algorithms. These steps will need to be further improved in the future to enable the integration of neoantigen identification and neoantigen-based therapy into the clinic.

Both methods can be used for HLA class I and class II-restricted peptides, however, prediction algorithms for HLA-II peptide prediction are less reliable and less well established. This is mainly due to a lack of data on endosomal processing of HLA-II molecules (Nielsen et al., 2010) and the more complex structure of the peptide binding groove (which has open ends in contrast to the HLA-I molecules) together with the possibility to bind longer peptides (11-20mers) (Lundegaard et al., 2007). Nevertheless, also proteogenomic identification of HLA-II-bound peptides is challenging due to the low abundance of antigen presenting cells (APCs) expressing HLA-II in tumor samples.

The current bottleneck of both approaches, however, is the correct prediction/identification of immunogenic neoantigens to select candidates for clinical applications. Therefore, researchers have tried to define several parameters that might be indicative of the immunogenicity of a given neoantigen candidate such as the above-mentioned DAI, the size of the peptide, the amino acid composition and the peptide-HLA complex stability (e.g. NetMHCstab) (Bräunlein et al., 2021; Calis et al., 2013; Garcia-Garijo et al., 2019; Jørgensen et al., 2014). The correct prediction of immunogenicity, however, remains largely impossible and several *in vitro* methods for immunogenicity assessment have been developed and deployed over the past years. These include the generation of large cDNA libraries of potential neoantigen genes that are transfected into target cells leading to their cell-intrinsic possessing and presentation towards T cells (Coulie et al., 1995; Huang et al., 2004). Another approach is the use of fluorescently labeled pHLA multimers/tetramers that will be bound by reactive T cells. By that, these reactive cells are labeled and can be sorted using florescence activated cell sorting (FACS) (Cohen et al., 2015; McGranahan et al., 2016), a method also applicable for high-throughput screening. Furthermore, syntenic peptides can be used for pulsing of HLA-matched APCs that activate potential immunogenic T cells using *in vitro* stimulations such as accelerated cocultured DC assays (acDC)(Bassani-Sternberg et al., 2016; Martinuzzi et al., 2011a) and that can be identified via interferon (IFN)-γ secretion. This approach can also be performed using peptide pools for a more high-throughput screening. The generation and transfection of several linked minigenes for each neoantigen candidate (tandem minigenes) into APCs for stimulation assays is also a used high-throughput screening method (Lu et al., 2014; Tran et al., 2015). All methods have successfully led to the identification of immunogenic neoantigens from tumor samples but still come along with their own limitations, which is mainly the presence and the viability of the respective neoantigen-reactive T cell in the used patient material. Therefore, the absence of an immunogenic T cell in such assays is no evidence for the lack of presence and potential immunogenicity in general.

As highlighted here, several methods for the identification of neoantigens and most importantly their immunological characterization have been developed and are currently in use, each having their own potentials and limitations. Overcoming these limitations for a more rapid and precise neoantigen identification is of great importance in the future for the implementation of neoantigen-based therapies into the clinic, of course also in a pan cancer setting.

1.3 Aim of this study

Within the Krackhardt lab it was previously shown as one of the first groups that cancer neoantigens can be directly identified from fresh tumor tissue using the above described proteogenomics approach (Bassani-Sternberg et al., 2016). However, the number of identified neoantigens per patient was limited (in total 11 neoantigens in 3 of 5 analyzed patients), as only mutations from DNA sequencing were used, and analysis focused on melanoma patients only, with a small cohort size of 5 patients.

In this study, the aim was to expand this analysis towards a pan-cancer cohort and evaluate if neoantigen discovery and immunogenicity assessment was feasible in a broader variety of cancer types and if tissue-agnostics biomarkers and potential common features and targets could be identified using a multi-omics data set.

Therefore, the primary objective was to establish an ImmuNEO MASTER pan-cancer cohort comprising of patients with various tumor entities, predominantly overlapping with the previously described MASTER cohort (Horak et al., 2021). Fresh tumor tissue as well as PBMCs from several timepoints were collected and a multi-omcis data set was generated using genomics, transcriptomics, immunopeptidomics and tumor microenvironment characterization together with several cooperation partners.

In a next step, improvements to the neoantigen identification pipeline were needed to be established within this thesis to broaden and enhance the potential for neoantigen identification towards non-coding regions and a bigger variety of variants. A comprehensive post-processing pipeline for the prioritization of neoantigens for immunogenicity assessment was needed to be developed and implemented as well.

After neoantigen identification, the scope of this thesis was to characterize the found aberrant peptides and evaluate their tumor-immunogenicity using patient derived PBMCs and TILs and potentially allogenic healthy donor PBMCs. Furthermore, reactive T cell clones should be isolated for a potential evaluation as T cell-based therapy.

To better understand the interplay between neoantigens and the tumor microenvironment and to potentially identify biomarkers related to neoantigen discovery, another aim was to characterize the tumor microenvironment using different methods and correlate findings with each other.

Finally, the multi-omics data comprising of genomics, transcriptomics, immunopeptidomics and tumor immunomics were thought to be integrated for the identification of tissue-agnostics biomarkers and potential common features and targets together with the patients' clinical data.

In conclusion, this thesis aims at identifying and characterizing neoantigens in a pan-cancer cohort, evaluating their potential as targets for immunotherapy and identifying other possible tissue-agnostic biomarkers that might improve immunotherapy on a systemic level.

2. Material and Methods

2.1 Material

2.1.1 Technical equipment

Table 4: List of technical equipment used for general lab work, cell culture and specific experiments.

Device	Company
Analytical balance SI-64	Denver Instrument / Sartorius AG, Göttingen, Germany
APOLLO Liquid nitrogen vacuum container	Cryotherm, Kirchen/Sieg, Germany
Autoclave Systec V95	Systec GmbH, Linden, Germany
BD™ LSR II	BD Biosciences, Franklin Lakes, USA
Biometra Mitsubishi P95 Printer	Biometra GmbH, Göttingen, Germany
BIOSAFE MD sample container	Cryotherm, Kirchen/Sieg, Germany
Centrifuge 5417R	Eppendorf AG, Hamburg, Germany
Centrifuge 5417R	Eppendorf AG, Hamburg, Germany
Centrifuge 5810R	Eppendorf AG, Hamburg, Germany
Centrifuge with vortex 7-0040	neoLab Migge GmbH, Heidelberg, Germany
Centrifuge with vortex 7-0040	neoLab Migge GmbH, Heidelberg, Germany
Digital microtiter shaker MTS 2/4	IKA®-Werke GmbH & CO. KG, Staufen, Germay
DynaMag™-2 Magnet	Invitrogen Dynal AS, Oslo, Norway
EcoVac Vacuum Pump	schuett-biotec GmbH, Göttingen, Germany
FACSAria III	BD Biosciences, Franklin Lakes, USA
Fume cupboard 2-453-DXNN	Köttermann GmbH & Co KG, Uetze/Hänigsen, Germany
HERAfreeze™ BASIC -86°C Freezer	Thermo Fisher scientific, Waltham, USA
ImmunoSpot S6 Ultra-V Analyzer	CTL - Europe GmbH, Bonn, Germany
Incubator BBD 6220	Heraeus Holding GmbH, Hanau, Germany
Incubator CB 150	BINDER GmbH, Tuttlingen, Germany
Irradiation chamber Cs137 Type Ob 29/902-1	Buchler GmbH, Braunschweig, Germany
Laminar flow HERAsafe KS 15	Heraeus Holding GmbH, Hanau, Germany
LS6000 sample container	tec-lab GmbH, Taunusstein, Germany
MACS MultiStand	Miltenyi Biotec GmbH, Bergisch Gladbach, Germany
Magnetic stirrer RH basic 2	IKA®-Werke GmbH & CO. KG, Staufen, Germay
Microscope Axiovert 40 C	Carl Zeiss AG, Feldbach, Schweiz
MidiMACS Separator	Miltenyi Biotec GmbH, Bergisch Gladbach, Germany
Minishaker MS2	IKA®-Werke GmbH & CO. KG, Staufen, Germay
Multichannel pipets	Eppendorf AG, Hamburg, Germany
Multifuge 3 S-R	Heraeus Holding GmbH, Hanau, Germany
Multifuge 3s	Heraeus Holding GmbH, Hanau, Germany
NALGENE Cryo 1°C Freezing Container	Thermo Fisher scientific, Waltham, USA
Neubauer improved counting chamber	Karl Hecht GmbH & Co KG, Sondheim/Röhn, Deutschland
OctoMACS Separator	Miltenyi Biotec GmbH, Bergisch Gladbach, Germany

Device	Company
Pipets	Eppendorf AG, Hamburg, Germany
Pipette controller	INTEGRA Biosciences GmbH, Biebertal, Germany
Precision balance 440	KERN & SOHN GmbH, Balingen, Germany
Premium -20°C Freezer	Liebherr-International Deutschland GmbH, Biberach an der Riß, Germany
Refrigerator Profi line	Liebherr-International Deutschland GmbH, Biberach an der Riß, Germany
Rotina 420R	Andreas Hettich GmbH & Co.KG, Tuttlingen, Germany
SunriseTM absorbance reader	Tecan Group Ltd., Männedorf, Switzerland
Thermomixer Compact	Eppendorf AG, Hamburg, Germany
UV Transilluminator	Biometra GmbH, Göttingen, Germany
Vortex Mixer 7-2020	neoLab Migge GmbH, Heidelberg, Germany
Vortexer Reax top	Heidolph Instruments GmbH & Co.KG, Schwabach, Germany
Vortex-Genie 2	Scientific Industries, Inc., New York, USA
VWR Power Source 300V	VWR International GmbH, Darmstadt, Germany
Waterbath	Memmert GmbH + Co. KG, Schwabach, Germany
Ziegra Ice machine	ZIEGRA Eismaschinen GmbH, Isernhagen, Germany

2.1.2 Consumables

Table 5: List of consumables used for general lab work, cell culture and specific experiments.

Consumable	Company
neoScrew Micro tubes 1.5ml brown	neoLab Migge GmbH, Heidelberg, Germany
Cell culture flask (T25, T75, T175)	Greiner Bio-One GmbH, Frickenhausen, Germany
Cell scraper	TPP Techno Plastic Products AG, Trasadingen, Schweiz
CyroPure tubes	Sarstedt AG & Co., Nümbrecht, Germany
Corning™ Falcon™ Round-Bottom Test Tubes with Cell Strainer Cap, 5mL	Corning, New York, USA
EIA/RIA plates	Corning, New York, USA
Falcon® Cell Strainer (100 adn 40 µm)	Corning, New York, USA
Falcons (15ml, 50 ml)	BD Biosciences, Franklin Lakes, USA
Filcon 30 µm filter	Syntec International, Dublin, Ireland
Gloves Dermatril P	KCL GmbH, Eichenzell, Germany
LD/LS columns	Miltenyi Biotec GmbH, Bergisch Gladbach, Germany
MAHAS4510 MultiScreen-HA 0.45 µm ELIspot plate	Merck KGaA, Darmstadt, Germany
Microtubes (1.2 ml)	Alpha Laboratories, Hampshire, UK
Nitrile gloves	Abena A/Sm Aabenraa, Denmark
Non-tissue culture treated plates (6-/24-well)	BD Biosciences, Franklin Lakes, USA
Nunc™ Cell culture flask (80cm2)	Thermo Fisher scientific, Waltham, USA
Parafilm M® laboratory film	Pechiney Plastic Packaging, Chicago, USA
PCR reaction tubes (0.5 ml)	VWR International GmbH, Darmstadt, Germany
Pipet tips (10/20/300/1250 µl)	Sarstedt AG & Co., Nümbrecht, Germany

Consumable	Company
QIAshredder Homogenizer	QIAGEN GmbH, Hilden, Germany
Reaction tubes (1.5, 2 ml)	Sarstedt AG & Co., Nümbrecht, Germany
Screw Cap Micro Tubes	Sarstedt AG & Co., Nümbrecht, Germany
Sealing foil (ELISA)	Alpha Laboratories, Hampshire, UK
Serological Pipets (5 ml, 10 ml, 25 ml, 50 ml)	Sarstedt AG & Co., Nümbrecht, Germany
Stericup/Steritop 0.22 µm filters	Merck KGaA, Darmstadt, Germany
Syringe filters (0.2, 0.45 µm)	TPP Techno Plastic Products AG, Trasadingen, Schweiz
Tissue culture-treated plates (48-well)	BD Biosciences, Franklin Lakes, USA
Tissue culture-treated plates (6-/12-/24-well, round/flat bottom 96-well)	TPP Techno Plastic Products AG, Trasadingen, Schweiz

2.1.3 Primary human material

All healthy volunteers and participating patients provided written informed consent in accordance with the local review board of the Ethics Commission of the Medical Faculty of the Technical University Munich and Ethics Committee of the Medical Faculty of the Heidelberg University as well as the principles of the Helsinki declaration. Information of analyzed patients and their HLA types is listed in Table 6, detailed clinical information is provided in Appendix 6.1 and 6.2. HLA types of healthy donors are listed in Table 7 and modified primary healthy donor cells are listed in Table 8.

Table 6: List of patients included in the ImmuNEO MASTER study and their HLA-types.
For patients marked with an asterisk, HLA typing was performed using targeted sequencing of PBMCs, for all other patients the HLA-type was determined by the consensus of three different typing algorithms using the DNA sequencing data (see 2.2.3.3 for methodological information).

Patient ID	HLA-A		HLA-B		HLA-C	
	(1)	(2)	(1)	(2)	(1)	(2)
ImmuNEO-01*	HLA-A0201	HLA-A2402	HLA-B4402	HLA-B4402	HLA-C0501	HLA-C0704
ImmuNEO-02	HLA-A0201	HLA-A1101	HLA-B3501	HLA-B4002	HLA-C0202	HLA-C0401
ImmuNEO-03	HLA-A0101	HLA-A2403	HLA-B3701	HLA-B5101	HLA-C0602	HLA-C1402
ImmuNEO-04*	HLA-A2301	HLA-A2402	HLA-B1501	HLA-B3801	HLA-C0602	HLA-C1203
ImmuNEO-05	HLA-A0301	HLA-A6601	HLA-B3701	HLA-B4102	HLA-C0602	HLA-C1703
ImmuNEO-08	HLA-A0301	HLA-A2601	HLA-B0702	HLA-B2705	HLA-C0102	HLA-C0702
ImmuNEO-09	HLA-A1101	HLA-A3201	HLA-B3503	HLA-B4006	HLA-C0401	HLA-C1502
ImmuNEO-11	HLA-A0201	HLA-A0301	HLA-B2705	HLA-B5201	HLA-C0102	HLA-C1202
ImmuNEO-13	HLA-A1101	HLA-A3101	HLA-B0801	HLA-B4402	HLA-C0501	HLA-C0701
ImmuNEO-14	HLA-A0301	HLA-A0301	HLA-B1801	HLA-B3801	HLA-C1203	HLA-C1203
ImmuNEO-15	HLA-A0201	HLA-A0301	HLA-B1501	HLA-B3701	HLA-C0401	HLA-C0602
ImmuNEO-16	HLA-A0301	HLA-A2301	HLA-B4001	HLA-B4901	HLA-C0304	HLA-C0701
ImmuNEO-17	HLA-A0201	HLA-A6802	HLA-B1402	HLA-B3906	HLA-C0702	HLA-C0802
ImmuNEO-18	HLA-A0301	HLA-A1101	HLA-B3501	HLA-B5701	HLA-C0401	HLA-C0602
ImmuNEO-19*	HLA-A0101	HLA-A2902	HLA-B3502	HLA-B4403	HLA-C0401	HLA-C1601
ImmuNEO-20	HLA-A0201	HLA-A0201	HLA-B5101	HLA-B5701	HLA-C0102	HLA-C0602
ImmuNEO-22*	HLA-A2902	HLA-A3201	HLA-B4402	HLA-B4403	HLA-C0501	HLA-C1601

Patient ID	HLA-A		HLA-B		HLA-C	
	(1)	(2)	(1)	(2)	(1)	(2)
ImmuNEO-23	HLA-A0301	HLA-A1101	HLA-B0702	HLA-B1803	HLA-C0701	HLA-C0702
ImmuNEO-24	HLA-A0206	HLA-A1101	HLA-B1525	HLA-B2704	HLA-C0702	HLA-C1202
ImmuNEO-25	HLA-A0101	HLA-A0301	HLA-B1801	HLA-B5101	HLA-C0701	HLA-C0602
ImmuNEO-26	HLA-A0201	HLA-A0201	HLA-B2705	HLA-B4402	HLA-C0202	HLA-C0501
ImmuNEO-27	HLA-A0101	HLA-A3301	HLA-B1402	HLA-B4002	HLA-C0202	HLA-C0802
ImmuNEO-28	HLA-A2301	HLA-A3001	HLA-B0702	HLA-B4403	HLA-C0401	HLA-C1203
ImmuNEO-30	HLA-A0101	HLA-A2601	n/a	HLA-B3701	n/a	HLA-C0602
ImmuNEO-31	HLA-A2902	HLA-A3201	HLA-B4002	HLA-B4501	HLA-C0202	HLA-C0602
ImmuNEO-32	HLA-A0101	HLA-A1101	HLA-B0702	HLA-B5601	HLA-C0102	HLA-C0702
ImmuNEO-33	HLA-A0201	HLA-A2601	HLA-B4102	HLA-B5201	HLA-C1202	HLA-C1701
ImmuNEO-34	HLA-A0101	HLA-A3201	HLA-B0801	HLA-B4002	HLA-C0202	HLA-C0701
ImmuNEO-35	HLA-A0201	HLA-A0301	HLA-B0702	HLA-B4001	HLA-C0304	HLA-C0702
ImmuNEO-36	HLA-A0201	HLA-A0301	HLA-B1402	HLA-B3503	HLA-C0401	HLA-C0802
ImmuNEO-37	HLA-A0301	HLA-A6801	HLA-B1801	HLA-B5101	HLA-C0701	HLA-C1504
ImmuNEO-38	HLA-A1101	HLA-A2601	HLA-B0702	HLA-B5101	HLA-C0102	HLA-C0702

Table 7: List of healthy donors used in this study and their HLA class I types.

Healthy donor	HLA-A	HLA-B	HLA-C
HD01	02:01	15:01, 44:02	03:03, 05:01
HD02	02:01	15:01, 47:02	04:01, 06:02
GS1	02:01, 23:01	07:02, 44:02	n/a
HD03	01:01, 03:01	07:02, 08:01	07:01, 07:02
HD04	01:01, 03:01	08:01, 56:01	01:02, 07:01
HD05	26:01, 03:01	03:02, 38:01	02:02, 12:03
HD06	02:01, 33:01	14:02, 51:01	08:02, 14:02
HD07	02:01, 26:01	44:02, 56:01	01:02, 05:01
HD08	02:01	07:02, 15:01	n/a
HD09	02:01, 03:01	07:01, 50:01	n/a

Table 8: List of modified primary human cells from healthy donors used for stimulation assays.

Primary cells	Characteristics	Source/Origin
HD04-18.2	PBMC transduced with KIF2C-specific TCR	Generated by Florian Dreyer

2.1.4 Cell lines

Table 9: List of standard and modified cell lines used for several different experiments.

Cell lines	Characteristics	Source/Origin
B95-8	Primate cell line infected with EBV	Ulrike Protzer, München
C1R	Human plasma leukemia cell line	Stefan Stevanovic, Tübingen
C1R-A6601	C1R, transduced with HLA-A*66:01-P2A-eGFP	[1]
C1R-B0702	C1R, transduced with HLA-B*07:02-P2A-eGFP	[1]
T2	T-cell leukemia/B-cell hybridoma; TAP-deficient	ATCC, Manassas, USA
T2-A0301	T2, transduced with HLA-A*03:01-P2A-eGFP	[1]
T2-B1501	T2, transduced with HLA-B*15:01-P2A-eGFP	[1]
T2-B4402	T2, transduced with HLA-B*44:02-P2A-eGFP	[1]

Table 10: List of commercial and self-made lymphoblastoid cell lines (see Methods 2.2.1.3) and their HLA class I types used as target cells in stimulation assays.

LCL	HLA-A*	HLA-B*	HLA-C*	IHW[2] number
LCL CLA	02:06,24:02	08:01,35	07	IHW09209
LCL Daudi	01:02,66:01	35:01,58:01	03:02,06:02	IHW09366
LCL FM	02:01, 24:02	07:02: 37:01	n/a	[3]
LCL HD04	01:01, 03:01	08:01, 56:01	01:02, 07:01	[4]
LCL HD06	02:01, 33:01	14:02, 51:01	08:02, 14:02	[4]
LCL HD07	02:01, 26:01	44:02, 56:01	01:02, 05:01	[4]
LCL HD08	02:01	07:02, 15:01	n/a	[4]
LCL IBW9	33:01	14:02	08:02	IHW09049
LCL IN-01	02:01, 24:02	44:02	05:01, 07:04	[4]
LCL IN-03	01:01, 24:03	37:01, 51:01	06:02, 14:02	[4]
LCL IN-04	23:01, 24:02	15:01, 38:01	06:02, 12:03	[4]
LCL IN-08	03:01, 26:01	07:02, 27:05	01:02, 07:02	[4]
LCL IN-09	11:01, 32:01	35:03, 40:06	04:01, 15:02	[4]
LCL IN-11	02:01, 03:01	27:05, 52:01	01:02, 12:02	[4]
LCL IN-13	11:01, 31:01	08:01, 44:02	05:01, 07:01	[4]
LCL IN-18	03:01, 11:01	35:01, 57:01	04:01, 06:02	[4]
LCL IN-19	01:01, 29:02	35:02, 44:03	04:01, 16:01	[4]
LCL IN-22	29:02, 32:01	44:02, 44:03	05:01, 16:01	[4]
LCL IN-24	02:06, 11:01	15:25, 27:04	07:02, 12:02	[4]
LCL IN-33	02:01, 26:01	41:02, 52:01	12:02, 17:01	[4]
LCL IN-37	03:01, 68:01	18:01, 51:01	07:01, 15:04	[4]
LCL Mel15	03:01, 68:01	27:05, 35:03	02:02, 04:01	[4]
LCL RSH	68:02,30:01	42:01	17:01	IHW09021
LCL SWEIG007	29:02	40:02	02:02	IHW09037

[1] These cell lines were retrovirally transduced and cloned jointly by members of Prof. Krackhardt's group.

[2] International HLA Workshop

[3] LCL previously generated by members of AG Krackhardt

[4] LCLs were generated according 2.2.1.3 from healthy donor or patient PBMCs within this study

2.1.5 Reagents and chemicals

Table 11: List of reagents and chemicals used for general lab work, cell culture and specific experiments.

Reagent/Chemical	Company
6x loading buffer	Thermo Fisher scientific, Waltham, USA
3-Amino-9-ethylcarbazole (AEC) tablets	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
7-Aminoactinomycin D (7-AAD)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Ampicillin	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
AIM V™	Thermo Fisher scientific, Waltham, USA
Bovine Serum Albumine (BSA)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Cyclosporin A	Klinikum rechts der Isar, TUM, Germany
DEPC H_2O	Thermo Fisher scientific, Waltham, USA
Dimethyl sulfoxide (DMSO)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
DNase I type IV	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Ethanol	Merck KGaA, Darmstadt, Germany
Ethidium monoazide bromide (EMA)	Thermo Fisher scientific, Waltham, USA
Fetal calf serum (FCS)	Thermo Fisher scientific, Waltham, USA
Ficoll	Biochrom GmbH, Berlin, Germany
Gentamycin	Biochrom GmbH, Berlin, Germany
HEPES	Thermo Fisher scientific, Waltham, USA
Hyaluronidase	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Hydrogen Peroxide Solution	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Human serum (HS)	Technische Universität München, Germany
Ionomycin	Merck KGaA, Darmstadt, Germany
Isopropanol	Merck KGaA, Darmstadt, Germany
L-Glutamine	Thermo Fisher scientific, Waltham, USA
Milk powder	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Non-essential amio acids (NEAA)	Thermo Fisher scientific, Waltham, USA
Paraformaldehyde (PFA)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Phosphate-buffered saline (PBS)	Thermo Fisher scientific, Waltham, USA
PBS powder without Ca^{2+}, Mg^{2+}	Merck KGaA, Darmstadt, Germany
Penicilline/Streptomycin	Thermo Fisher scientific, Waltham, USA
Phorbol 12-myristate 13-acetate (PMA)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Protamine Sulfate	MP Biomedicals GmbH, Illkirch, France
RNA protect	QIAGEN GmbH, Hilden, Germany
RPMI-1640	Thermo Fisher scientific, Waltham, USA
Sodium carbonate (Na_2CO_3)	Merck KGaA, Darmstadt, Germany
Sodium hydrogen carbonate ($NaHCO_3$)	Merck KGaA, Darmstadt, Germany
Sodium Pyruvate	Thermo Fisher scientific, Waltham, USA
Streptavidin-horseradish peroxidase (HRP)	Mabtech AB, Nacka Strand, Sweden
Sulfuric acid	Carl Roth GmbH + Co. KG, Karlsruhe, Germany
TRIzol reagent	Thermo Fisher scientific, Waltham, USA
Trypane blue	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
Trypsine/EDTA	Thermo Fisher scientific, Waltham, USA
Tween 20	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany

2.1.6 Kits

Table 12: List of kits used for several different experiments.

Kit	Purpose	Company
AEC substrate set	ELIspot development	BD Biosciences, Franklin Lakes, USA
BD OptEIA™ Human IFN-γ ELISA Set	Cytokine measurement in cell culture supernatants	BD Biosciences, Franklin Lakes, USA
CD137 MicroBead Kit	T cell enrichment of neoantigen-activated cells	Miltenyi Biotec GmbH, Bergisch Gladbach, Germany
Tumor dissociation kit	Tumor tissue digestion	Miltenyi Biotec GmbH, Bergisch Gladbach, Germany
Venor GeM mycoplasma detection kit	Testing of cell lines for absence of mycoplasma infection	Minerva Biolabs GmbH, Berlin, Germany

2.1.7 Media and buffer

Table 13: List of buffers used for general lab work, cell culture and specific experiments.

Buffer/solution	Application	Ingredients
AEC buffer	ELIspot	100 µl AEC solution + 1 drop chromogen (AEC substrate set kit)
Blocking solution	ELISA	PBS + 1% (w/v) milk powder
ΔFCS	Multiple applications	FCS, inactivated for 20 min at 58°C
ΔHS	Multiple applications	HS, inactivated for 20 min at 58°C
ELISA coating buffer	ELISA	H_2O + 0.1 mol/l $NaHCO_3$, 0.03 mol/l Na_2CO_3, pH = 9.5
FACS buffer	Stainings for flow cytometry	PBS + 1% ΔFCS
HRP-complex solution	ELIspot	10ml PBS + 50 µl von Strp. / HRP + 50 µl ΔFCS
Washing buffer	ELIspot, ELISA	PBS + 0.05% v/v Tween 20

Table 14: List of media used for general lab work, cell culture and specific experiments.

Medium	Ingredients
AIM-V	AIM-V (Thermo Fisher scientific), no supplements
cRPMI	RPMI supplemented with 10% ΔFCS, 10 mM non-essential amino acids, 1 mM sodium pyruvate, 2 mM L-Glutamine, 100 U/ml Penicillin and 100 µg/ml Streptomycin
Freezing medium	90% ΔFCS + 10% DMSO
T-cell medium (TCM)	RPMI 1640 supplemented with 5% v/v ΔFCS, 5% ΔHS, 10 mM non-essential amino acids, 1 mM sodium pyruvate, 2 mM L-Glutamine, 100 U/ml Penicillin, 100 µg/ml Streptomycin, 10 mM HEPES buffer and 16.6 µg/ml Gentamycin
Tumor digestion medium	RPMI supplemented with 0.25 mg/ml DNase I, 0.25 mg/ml Hyaluronidase, Enzyme A (1:250, Tumor dissociation kit, Miltenyi) and Enzyme H (1:25, Tumor dissociation kit, Miltenyi)

2.1.8 Recombinant cytokines and TLR ligands

Table 15: List of recombinant human (rh) cytokines and toll-like receptor (TLR) ligands in cell culture and specific experiments.

Substance	Company
OKT-3	Kindly provided by Elisabeth Kremmer, Helmholtz Zentrum München
rh GM-CSF	PeproTech, London, UK
rh IFN-g	PeproTech, London, UK
rh IL-15	PeproTech, London, UK
rh IL-1b	PeproTech, London, UK
rh IL-2	PeproTech, London, UK
rh IL-4	PeproTech, London, UK
rh IL-7	PeproTech, London, UK
rh TNF-a	PeproTech, London, UK

2.1.9 Peptides

Table 16: List of synthetic mutated peptides used for stimulation assays.

Peptide ID	Sequence	Company
IN_01_A	ALSGHLET*L*	DGPeptides Co. Ltd, Hangzhou, China
IN_01_B	KGDSPQVKLKY	DGPeptides Co. Ltd, Hangzhou, China
IN_01_C	GHPSGARAM	DGPeptides Co. Ltd, Hangzhou, China
IN_01_D	KELCKQIQL	DGPeptides Co. Ltd, Hangzhou, China
IN_02_A	TGGQKYRTK	DGPeptides Co. Ltd, Hangzhou, China
IN_03_A	A*A*SASRVQVI	DGPeptides Co. Ltd, Hangzhou, China
IN_03_B	VDS*R*GSLF	DGPeptides Co. Ltd, Hangzhou, China
IN_03_C	ESKDFCVM	DGPeptides Co. Ltd, Hangzhou, China
IN_03_D	G*S*HDQAMHF	DGPeptides Co. Ltd, Hangzhou, China
IN_03_E	TDGGGRAKL	DGPeptides Co. Ltd, Hangzhou, China
IN_03_F	TFQ*K*KTKEM	DGPeptides Co. Ltd, Hangzhou, China
IN_04_A	AGVVLGGL	DGPeptides Co. Ltd, Hangzhou, China
IN_04_B	FLLLLLKNF	DGPeptides Co. Ltd, Hangzhou, China
IN_04_C	GL*A*ATFASL	DGPeptides Co. Ltd, Hangzhou, China
IN_04_D	KTKEM*S*NNVK	DGPeptides Co. Ltd, Hangzhou, China
IN_04_E	LGG*T*GASF	DGPeptides Co. Ltd, Hangzhou, China
IN_04_F	NTLMSLSDM	DGPeptides Co. Ltd, Hangzhou, China
IN_04_G	SYLSNISY	DGPeptides Co. Ltd, Hangzhou, China
IN_04_H	TSLA*A*NTF	DGPeptides Co. Ltd, Hangzhou, China
IN_04_I	*T*VHSTSIAF	DGPeptides Co. Ltd, Hangzhou, China
IN_04_J	GHGQPWNSL	DGPeptides Co. Ltd, Hangzhou, China
IN_04_K	**HAGAALHLH	DGPeptides Co. Ltd, Hangzhou, China
IN_04_L	KLQNA*S*KKLF	DGPeptides Co. Ltd, Hangzhou, China
IN_04_M	KSAGI*A*GL	DGPeptides Co. Ltd, Hangzhou, China
IN_05_A	DIFSRISQ*R*	DGPeptides Co. Ltd, Hangzhou, China
IN_05_B	E*T*NKSLLKR	DGPeptides Co. Ltd, Hangzhou, China
IN_05_C	DLLEPG*G*QR	DGPeptides Co. Ltd, Hangzhou, China
IN_05_D	SL*G*AGRWRL	DGPeptides Co. Ltd, Hangzhou, China

Peptide ID	Sequence	Company
IN_08_A	LSELDVSVR	DGPeptides Co. Ltd, Hangzhou, China
IN_08_B	PQESAPAAL	DGPeptides Co. Ltd, Hangzhou, China
IN_08_C	APVLKS*A*R	DGPeptides Co. Ltd, Hangzhou, China
IN_08_D	GLEPGKCSP	DGPeptides Co. Ltd, Hangzhou, China
IN_08_E	GPLGPR*G*SI	DGPeptides Co. Ltd, Hangzhou, China
IN_08_F	NRITEVSAK	DGPeptides Co. Ltd, Hangzhou, China
IN_08_G	SAGAAAQGRAGGAP	DGPeptides Co. Ltd, Hangzhou, China
IN_08_H	TQAL*V*LAPTQ	DGPeptides Co. Ltd, Hangzhou, China
IN_11_A	SAAEL*H*HV	DGPeptides Co. Ltd, Hangzhou, China
IN_11_B	GGITAVT*L*N	DGPeptides Co. Ltd, Hangzhou, China
IN_11_C	**RGISWRSHL	DGPeptides Co. Ltd, Hangzhou, China
IN_11_D	S*R*SVAQAGVQR	DGPeptides Co. Ltd, Hangzhou, China
IN_11_E	**VAAGPGAV	DGPeptides Co. Ltd, Hangzhou, China
IN_13_A	KLPTLPKKY	DGPeptides Co. Ltd, Hangzhou, China
IN_13_B	LFKNLTIL	DGPeptides Co. Ltd, Hangzhou, China
IN_15_A	ICTT*S*VSK	DGPeptides Co. Ltd, Hangzhou, China
IN_15_B	LRAVTL*I*AK	DGPeptides Co. Ltd, Hangzhou, China
IN_17_A	MQSRLTA*A*	DGPeptides Co. Ltd, Hangzhou, China
IN_17_B	*A*GLSHHAL	DGPeptides Co. Ltd, Hangzhou, China
IN_18_A	MRL*W*SQLL	DGPeptides Co. Ltd, Hangzhou, China
IN_19_A	GRPGTRPAL	DGPeptides Co. Ltd, Hangzhou, China
IN_19_B	SES*N*VDRLM	DGPeptides Co. Ltd, Hangzhou, China
IN_19_C	ST*L*VLDEFKR	DGPeptides Co. Ltd, Hangzhou, China
IN_19_D	**VASISLTK	DGPeptides Co. Ltd, Hangzhou, China
IN_19_E	*G*SLNGGKPFLQAFY	DGPeptides Co. Ltd, Hangzhou, China
IN_19_F	KKY*W*VGAKL	DGPeptides Co. Ltd, Hangzhou, China
IN_19_G	KVGSLAG*F*	DGPeptides Co. Ltd, Hangzhou, China
IN_19_H	MPEHQSTAL	DGPeptides Co. Ltd, Hangzhou, China
IN_19_I	*R*RLQRDKIA	DGPeptides Co. Ltd, Hangzhou, China
IN_22_A	PPSEAQP*L*P	DGPeptides Co. Ltd, Hangzhou, China
IN_23_A	ASASQSA*G*IIGMSH	DGPeptides Co. Ltd, Hangzhou, China
IN_23_B	GAPAPVMVEK	DGPeptides Co. Ltd, Hangzhou, China
IN_24_A	S*R*VVGITGVP	DGPeptides Co. Ltd, Hangzhou, China
IN_24_B	LPIYGRAR	DGPeptides Co. Ltd, Hangzhou, China
IN_24_C	STMVKGRQTTTK	DGPeptides Co. Ltd, Hangzhou, China
IN_27_A	EGVAGPHSR	DGPeptides Co. Ltd, Hangzhou, China
IN_28_A	RVWD*V*SGLRKK	DGPeptides Co. Ltd, Hangzhou, China
IN_28_B	SP*R*QPPLLL	DGPeptides Co. Ltd, Hangzhou, China
IN_28_C	VIHPP*R*PPK	DGPeptides Co. Ltd, Hangzhou, China
IN_28_D	DTAPSGESR	DGPeptides Co. Ltd, Hangzhou, China
IN_28_E	E*P*LTTREI	DGPeptides Co. Ltd, Hangzhou, China
IN_28_F	GARLSSGRL	DGPeptides Co. Ltd, Hangzhou, China
IN_28_G	VGSGLGPGWVM	DGPeptides Co. Ltd, Hangzhou, China
IN_30_A	**QCKRSSSSYR	DGPeptides Co. Ltd, Hangzhou, China
IN_32_A	AP*K*SSSGFSL	DGPeptides Co. Ltd, Hangzhou, China

Peptide ID	Sequence	Company
IN_32_B	GP*G*SIQKR	DGPeptides Co. Ltd, Hangzhou, China
IN_32_C	ST*M*SALPNSR	DGPeptides Co. Ltd, Hangzhou, China
IN_33_A	EA*E*VEESLGLR	DGPeptides Co. Ltd, Hangzhou, China
IN_34_A	**SEVQDRAVP	DGPeptides Co. Ltd, Hangzhou, China
IN_36_A	AGLGGVKL	DGPeptides Co. Ltd, Hangzhou, China
IN_37_A	A*T*ERKEAK	DGPeptides Co. Ltd, Hangzhou, China
IN_37_B	DVVVVH*R*RR	DGPeptides Co. Ltd, Hangzhou, China
IN_37_C	G*S*PSLSQR	DGPeptides Co. Ltd, Hangzhou, China
IN_37_D	KFAQK*V*LR	DGPeptides Co. Ltd, Hangzhou, China
IN_37_E	RLANTQ*A*KKAK	DGPeptides Co. Ltd, Hangzhou, China
IN_37_F	SAADVVVVH*R*	DGPeptides Co. Ltd, Hangzhou, China
IN_37_G	TVG*V*PTVLEKLQK	DGPeptides Co. Ltd, Hangzhou, China
IN_37_H	VDAN*R*KIY	DGPeptides Co. Ltd, Hangzhou, China
IN_38_A	DVI*R*KALQY	DGPeptides Co. Ltd, Hangzhou, China
IN_38_B	RP*H*VGIHL	DGPeptides Co. Ltd, Hangzhou, China
IN_38_C	SITPGT*V*L	DGPeptides Co. Ltd, Hangzhou, China
IN_38_D	SQSTTASL*F*KK	DGPeptides Co. Ltd, Hangzhou, China
IN_38_E	STTASL*F*KK	DGPeptides Co. Ltd, Hangzhou, China
Mel15_KIF2C	RLF*L*GLAIK	DGPeptides Co. Ltd, Hangzhou, China

Table 17: List of wild-type peptides used as control peptides for stimulation assays.

Peptide ID	Sequence	Company
IN_01_wt	DAARRNSW	DGPeptides Co. Ltd, Hangzhou, China
IN_04_wt	GLTATFASL	DGPeptides Co. Ltd, Hangzhou, China
IN_11_wt	ISAAELRHV	DGPeptides Co. Ltd, Hangzhou, China
IN_38_wt	STTASLSKK	DGPeptides Co. Ltd, Hangzhou, China
Mel15_KIF2C_wt	RLFPGLAIK	DGPeptides Co. Ltd, Hangzhou, China

2.1.10 Antibodies

Table 18: List of antibodies used in flow-cytometry analysis for tumor microenvironment phenotyping.

Antibody	Clone	Conjugation	Company
anti-human CD103	Ber-ACT8	FITC	BD Biosciences, Franklin Lakes, USA
anti-human CD11b	ICRF44	PE, AF700	BD Biosciences, Franklin Lakes, USA
anti-human CD11c	S-HCL-3	PE	BioLegend Inc., San Diego, USA
anti-human CD127	A019D5	BV510	BioLegend Inc., San Diego, USA
anti-human CD14	MoP9	APC-H7	BD Biosciences, Franklin Lakes, USA
anti-human CD15	HI98	PerCP-Cy5.5	BD Biosciences, Franklin Lakes, USA
anti-human CD152	BNI3	PeCy5	BD Biosciences, Franklin Lakes, USA
anti-human CD16	3G8	PB	BD Biosciences, Franklin Lakes, USA
anti-human CD20	H1	BV510	BD Biosciences, Franklin Lakes, USA
anti-human CD25	2A3	PE	BD Biosciences, Franklin Lakes, USA
anit-human CD223 (LAG-3)	3DS223H	APC	Thermo Fisher, Waltham, USA
anti-human CD274 (PD-L1)	MIH1	APC-R700	BD Biosciences, Franklin Lakes, USA
anti-human CD279 (PD-1)	EH12.2H7	PeCy7	BioLegend Inc., San Diego, USA

Antibody	Clone	Conjugation	Company
anti-human CD3	UCHT1	AF700	BioLegend Inc., San Diego, USA
anti-human CD33	HIM3-4	FITC	BD Biosciences, Franklin Lakes, USA
anti-human CD33	WM53	V450	BD Biosciences, Franklin Lakes, USA
anti-human CD366 (TIM-3)	7D3	BB515	BD Biosciences, Franklin Lakes, USA
anti-human CD4	SK3	V450	BD Biosciences, Franklin Lakes, USA
anti-human CD45	2D1	APC-H7	BD Biosciences, Franklin Lakes, USA
anti-human CD45	HI30	PerCP-Cy5.5, V500	BD Biosciences, Franklin Lakes, USA
anti-human CD45RA	HI100	BV510	BioLegend Inc., San Diego, USA
anti-human CD47	CC2C6	PeCy7	BioLegend Inc., San Diego, USA
anti-human CD56	5.1H11	PE	BioLegend Inc., San Diego, USA
anti-human CD62L	DREG-56	PE	BD Biosciences, Franklin Lakes, USA
anti-human CD64	10.1	PeCy7	BD Biosciences, Franklin Lakes, USA
anti-human CD66b	G10F5	PeCy7	BioLegend Inc., San Diego, USA
anti-human CD73	AD2	APC	BioLegend Inc., San Diego, USA
anti-human CD8	SK1	APC-H7	BD Biosciences, Franklin Lakes, USA
anti-human EpCam	9C4	PE	BioLegend Inc., San Diego, USA
anti-human HLA-DR	G46-6	APC	BD Biosciences, Franklin Lakes, USA
anti-human Vimentin	RV202	AF488	BD Biosciences, Franklin Lakes, USA
Isotype IgG1	MOPC-21	AF488, AF700, APC-H7, PeCy7, PerCP-Cy5.5, FITC, PB, PE, V450	BD Biosciences, Franklin Lakes, USA
Isotype IgG1	MOPC-21	APC, BV510	BioLegend Inc., San Diego, USA
Isotype IgG1	X40	APC-R700, BB515, V500	BD Biosciences, Franklin Lakes, USA
Isotype IgG2a	G155-178	APC, PeCy5	BD Biosciences, Franklin Lakes, USA
Isotype IgG2b	27-35	APC-H7	BD Biosciences, Franklin Lakes, USA
Isotype IgG2b	MPC-11	BV510, PE	BioLegend Inc., San Diego, USA
Isotype IgM	G155-228	PerCP-Cy5.5	BD Biosciences, Franklin Lakes, USA

Table 19: List of antibodies used for fluorescence activated cell sorting (FACS) of different cell types in the tumor microenvironment of primary human tumor tissues.

Antibody	Clone	Conjugation	Company
anti-human CD33	WM53	V450	BD Biosciences, Franklin Lakes, USA
anti-human CD4	HI30	AF700	BD Biosciences, Franklin Lakes, USA
anti-human CD45	J33		Beckman CoulterGmbH, Krefeld, Germany
anti-human CD8	RPAT-8	PeCy7	BD Biosciences, Franklin Lakes, USA
anti-human EpCam	9C4	PE	BioLegend Inc., San Diego, USA
anti-human Vimentin	RV202	AF488	BD Biosciences, Franklin Lakes, USA

Table 20: List of antibodies used for ELIspot analysis.

Antibody	Clone	Conjugation	Company
anti-IFNγ coating antibody	1-D1K	None	Mabtech AB, Nacka Strand, Sweden
anti-IFNγ capture mAb	7-B6-1	biotin	Mabtech AB, Nacka Strand, Sweden

2.1.11 Software and web-based tools

Table 21: List of software tools.

Software	Application	Company
FlowJo v10.7.1	Flow cytometry analysis	Tree Star, Ashland, USA
Graphpad Prism v6	Data processing, analysis and plotting	GraphPad Software, Inc., La Jolla, USA
Illustrator	Plotting	Adobe Limited, Dublin, Ireland
Immunospot software 5.4.0.1	ELIspot analyses	CTL-Europe, Bonn, Germany
Mendeley	Citation management	Mendeley Ltd., London, England
MHCflurry 1.6.0	In-silico epitope prediction	n/a
Microsoft Office (Word, Excel, Powerpoint), 2010	Data processing and presentation	Microsoft Corporation, Redmond, USA
pFIND	MS spectra matching	n/a
R studio	Data processing, analysis and plotting	Prosit, Boston, USA

Table 22: List of web-based tools.

Tool	Application	Homepage
BioRender	Generation of plots	https://www.biorender.com
CTpedia	CTA gene selection	http://www.cta.Incc.br/
Ensembl GRCh38.92	Sequence extraction from reference genome	http://www.ensembl.org/index.html
Genotype-Tissue Expression (GTEx)	Human healthy tissue RNA sequencing data	https://gtexportal.org
Human Protein Atlas	Cancer-associated gene selection	http://www.proteinatlas.org/
IEDB	Neoantigen filtering	https://www.iedb.org/
MHCMotifDecon-1.0	MHC-peptide motif identification	https://services.healthtech.dtu.dk/services/MHCMotifDecon-1.0/
MsigDB v7.4	Hallmark and Gene Ontology gene set definitions	https://www.gsea-msigdb.org/gsea/msigdb/
NCBI Basic Local Alignment Search Tool (BLAST)	Neoantigen filtering	https://blast.ncbi.nlm.nih.gov/Blast.cgi
NetMHC 4.0	In-silico epitope prediction	http://www.cbs.dtu.dk/services/NetMHC/
PepBank	Neoantigen filtering	http://pepbank.mgh.harvard.edu/
PeptideAtlas	Neoantigen filtering	http://www.peptideatlas.org/
REDIportal	RNA editing data base	http://srv00.recas.ba.infn.it/atlas/
UCSC Genome Browser Gateway (BLAT)	Neoantigen post-processing, variant examination	http://genome-euro.ucsc.edu

2.2 Methods

2.2.1 Cell culture

Processing and cultivation of cells were carried out under sterile conditions. All procedures involving human blood samples, human primary cell lines and Epstein Barr virus (EBV)-transformed cells were performed according to S2 safety guidelines.

2.2.1.1 Processing of human material, isolation and cultivation of primary human cells

Informed consent was obtained from all participants in accordance with the requirements of the institutional review boards (Ethics Commission of the Medical Faculty of Technical University Munich and Ethics Committee of the Medical Faculty of Heidelberg University (S-206/2011)). An overview about all patients is provided in 6.1.

PBMC from both patients and healthy donors were isolated from whole blood immediately upon receipt using density-gradient centrifugation (Ficoll/Hypaque, Biochrom). Therefore, whole blood was transferred in a 50 ml falcon and filled up until 35 ml with RPMI or PBS. Each mixture was then carefully overlaid on 15 ml Ficoll solution in a fresh 50 ml falcon. After centrifugation for 25 min at 880 *g* without brake, the leukocyte layer was carefully extracted using a 10-ml serological pipet or a 1-ml pipet. Cell suspensions were washed once with RPMI or PBS and counted. Purified PBMC were then frozen for use in subsequent downstream applications.

Tumor tissue samples were obtained from patients who underwent tumor resection at various DKTK partner sites. Immediately post-resection, an experienced pathologist performed macroscopic dissection of fresh tumor tissue, which was then stored in PBS at 4°C for transportation or until processing. Remaining tumor tissue was then formalin-fixed and paraffin-embedded (FFPE). Prior to further analysis, a pathologist confirmed the tumor diagnosis, and the tumor content was assessed using an HE stain of the sample.

Parts of the fresh tumour tissue was snap frozen in liquid nitrogen (-196 °C) immediately on receipt and stored in a nitrogen tank until subsequent sequencing (see 2.2.3.1) and mass spectrometry analysis (see 2.2.4.1).

For the isolation and cultivation of TILs, part of the fresh tumor tissues was minced and tissue pieces were cultivated with γ-irradiated feeder cells (30 Gy, ^{137}CS) in TCM supplemented with 1000 units (U)/ml IL-2 and 30 ng/ml OKT3. A medium change, TCM supplemented with 300 U/ml IL-2, was carried out twice a week. Following a two-week expansion period, TILs were counted and frozen for subsequent use in stimulation assays (see 2.2.7.2).

The remaining tumor tissue was used for digestion and subsequent flow-cytometry analysis and sort (see 2.2.2.1 and 2.2.2.2).

2.2.1.2 Cultivation of cell lines

All cell lines used in this study were grown as suspension cell lines and cultivated in cRPMI. Growth behaviour and morphology of all cell lines was routinely monitored, with cells being subcultured twice a week or as needed when the medium color indicated high cell density. Regular confirmation of the absence of mycoplasma infection was conducted through polymerase chain reaction using the Venor GeM mycoplasma detection kit according to manufacturer's instructions.

2.2.1.3 Generation of lymphoblastoid cell lines

The generation of patient and healthy donor derived lymphoblastoid cell lines (LCL) was performed according to standard procedures using EBV for immortalization.

First, potent EBV-containing supernatant was generated from B95-8 cells. Therefore, B95-8 cells were expanded in cRPMI until sufficient cell numbers were reached. Then, 1 Mio cells per ml were stimulated with 20 ng/ml PMA in cRPMI for 1 h at 37°C, followed by three washes and reculturing at a density of 1 Mio cells per ml in fresh cRPMI. The supernatant was harvested after 3 days, filtered using a 0.45 µm sterile filter and stored for up to 1 year at -80°C.

In a second step, this EBV-containing supernatant was employed for the infection and immortalization of patient-derived B cells obtained from human primary PBMC samples (healthy donors and patients). Therefore, up to 5 Mio PBMCs were incubated in 1 ml cRPMI and 1 ml EBV supernatant for 2 h at 37°C. Subsequently, an additional 1 ml cRPMI supplemented with Cyclosporine A, resulting in a final concentration of 1 µg/ml, was added, and cells were cultured in T25 cell culture flasks at 37°C. Cells were split once clusters became visible and/or the medium colour changed. Expansion continued at a concentration of 0.3-0.6 Mio cells per ml until a sufficient quantity of cells was obtained for freezing or direct use in experiments. A list of all generated LCLs can be found in Table 10.

2.2.1.4 Freezing, thawing and counting of cells

For cryopreservation of viable cells, cells were pelleted by centrifugation and the supernatant was discarded. Cells were then resuspended in 1 ml freezing medium per aliquot and transferred into labeled cyro tubes. Cyro tubes were placed in a NALGENE Cryo 1°C Freezing Container at -80°C and transferred to a liquid nitrogen for long-term storage the following day.

For re-cultivation, 10 ml RPMI was pre-warmed to 36°C in a water bath. Frozen cell-media suspensions were rabidly thawed by adding small amounts of pre-warmed RPMI. Thawed material was transferred directly to the prepared falcon with 10 ml pre-warmed RPMI. Removal of residual DMSO was achieved through centrifugation at 500 *g* for 5 min and subsequent resuspension of the cell pellet in the appropriate cell culture medium.

Cell counting was performed using a Neubauer counting chamber. Therefore, the cell suspension was diluted with Trypane blue in ratio of 1:4 (or other ratios ranging from 1:2 to 1:10 to ensure a minimum of 3 cells per quarter). Cells were counted in 16 quarters of each corner and cell concentration was calculated using the following formula:

c_{cells} [Mio/ml] = [counter number of cells in all 4 corners]/4 * dilution factor * 10 000 / 10^6

2.2.2 Phenotyping of the tumor microenvironment

Phenotyping of the tumor microenvironment was performed using two approaches. First, several TIL populations, tumor cells and fibroblast together with the expression of regulatory surface marker were assessed using surface-based flow-cytometry analysis. Second, fluorescence activated cell sorting (FACS) was used to sort specific TIL populations (focusing on $CD8^+$ T cells) and the gene expression pattern of these cell types was analysis using bulk mRNA sequencing (RNA extraction, library preparation and sequencing performed by AG Klink, Dresden; data analysis performed in cooperation with AG Rad).

To enable the flow-cytometric analysis, first a single cell suspension was generated from the fresh human tumor tissue. Therefore, the tumor was minced and 0.2 g tissue pieces per tube was digested for 90 min at 37°C in 1 ml of tumor digestion medium (2.1.7). After digest the suspension and tissue pieces were meshed through a 100 µm mesh, pooled and counted. Single cells were used for flow-cytometry analysis and FACS.

2.2.2.1 Flow cytometry analysis of tumor single cell suspensions

For flow cytometry analysis, up to 0.5 Mio alive single cells from the digested tumor tissue was used per panel and isotype controls. Depending on the total number of available single cells after digestion, less cells per panel were used or not all panels were stained. For the staining process, cells were first incubated in 50 µl of human serum (HS) for 20 min to block unspecific binding. Ethidium monoazide (EMA) was subsequently added in a 1:500 ratio to the HS for live-dead staining. Cells were then incubated 10 min on ice in the dark and an additional 10 min on ice in the light. Following a wash with FACS buffer, 2 µl of the respective specific antibodies or 1.5 µl of the isotype control antibodies (see Table 18) were added according to the panels in Table 23. Appropriate isotype controls for each antibody were utilized as negative controls in two isotype control panels. Cells were stained for 20 min on ice in the dark, washed with FACS buffer, fixed with 1% paraformaldehyde (PFA) and stored at 4°C for later analysis. For single stains, cells stained with EMA only were used as well as anti-IgG beads stained with an antibody of each fluorophore according to manufacturers' instructions. All steps were conducted on ice and as fast as possible to minimize alterations in cell viability and marker expression. Measurements were performed on an LSR II and unstained cells as well as single stains were used for instrument set-up. Voltages were adjusted to the autofluorescence of each patient tumor and as many events as possible were measured using FACS DIVA software. Data analysis and compensation were performed using FlowJo V10.7.1, with a consistent gating strategy applied for every sample based on the respective panels (schematic gating strategy see Figure 4). Final data analysis of the exported FlowJo data was performed using a custom R script as shown in Appendix 6.9.6.

Table 23: Overview of all used flow-cytometry phenotyping panels.

Channel LSR II	Panel general	Panel 1.1	Panel 1.2	Panel 2.1	Panel 2.2
1	Vimentin-AF488	CD366-BB515	CD103-FITC	CD336-BB515	CD336-BB515
2	EpCam-PE	CD62L-PE	CD62L-PE	CD56-PE	CD25-PE
3	EMA	EMA	EMA	EMA	EMA
4	CD45-PerCP-Cy5.5	CD152-PeCy5	CD45-PerCP-Cy5.5	CD152-PeCy5	CD152-PeCy5
5	CD33-PeCy7	CD297-PeCy7		CD297-PeCy7	CD297-PeCy7
6	CD4-V450	CD4-V450	CD4-V450	CD33-V450	CD4-V450
7	CD19-BV510	CD45RA-BV510	CD45RA-BV510	CD19-BV510	CD127-BV510
8		CD223-APC	HLA-DR-APC	CD223-APC	CD223-APC
8.5	CD3-AF700	CD3-AF700	CD3-AF700	CD3-AF700	CD3-AF700
9	CD8-APC-H7	CD8-APC-H7	CD8-APC-H7	CD45-APC-H7	

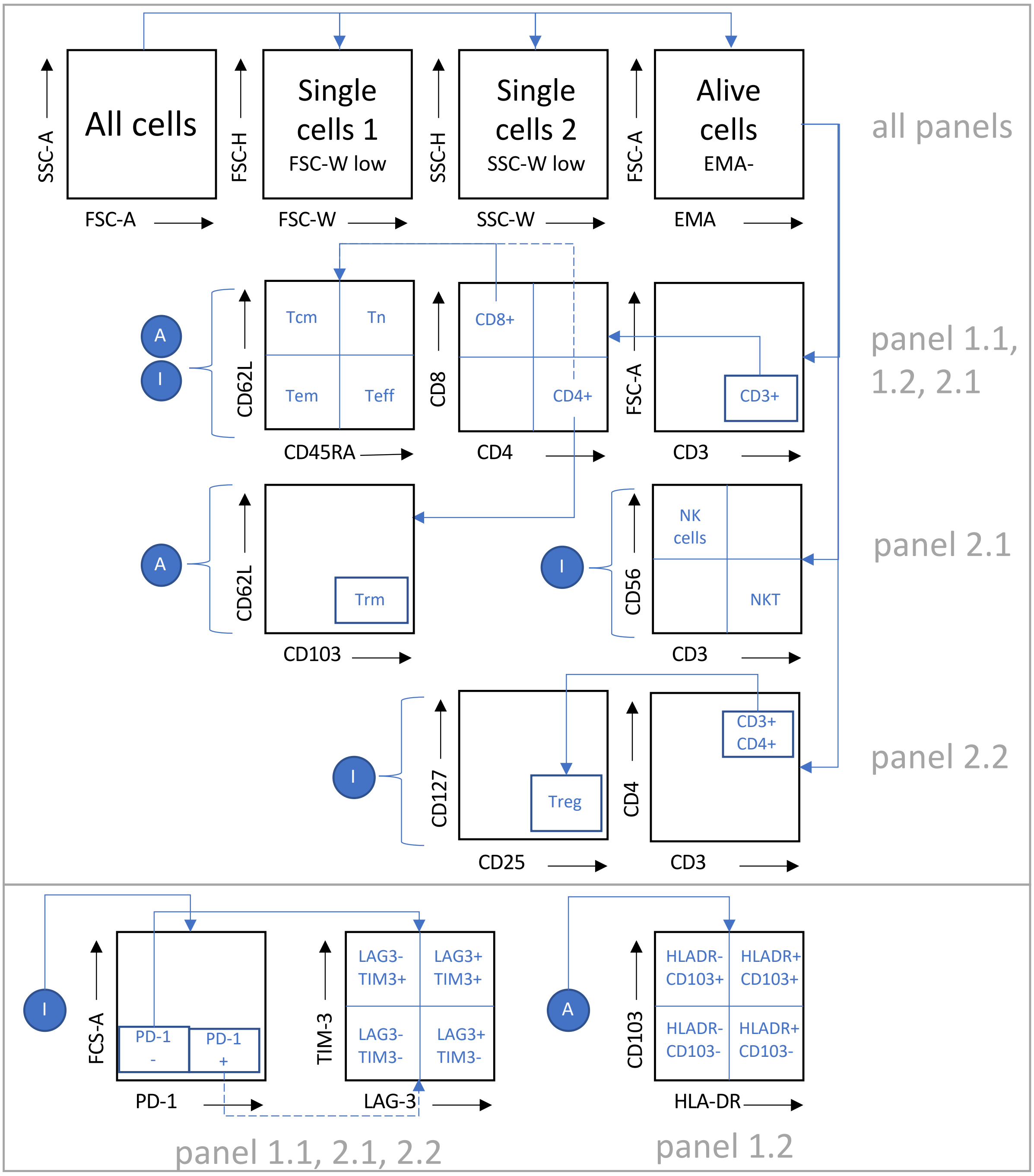

Figure 4: Schematic gating strategy for phenotyping of primary tumor-infiltrating lymphocytes using flow-cytometry.
For all panels, all cells according to size (FSC) and granularity (SSC) were selected excluding debris, single cells were selected using two different consecutive single cell gates and alive cells were gated as EMA negative. Following this general pre-gating, different gating strategies for each panel were used as indicated above. Black boxes illustrate the scatter plots labeled with the respective analyzed surface marker (for corresponding fluorophore refer to Table 23). Blue boxes show the approximate position of each used gate and blue text indicates the cell type gated for. Blue arrows indicate the next sub-gate applied to the gated population. For T cell subsets the expression of several inhibitory (labeled with I) and activation (labeled with A) marker was also analyzed as shown at the bottom. CD, cluster of differentiation; EMA, ethidium monoazide; FCS-A/W/H, forward scatter-area/width/height; HLA, human leukocyte antigen; LAG-3, lymphocyte-activation gene 3; NK, natural killer; NKT, natural killer T cells; SSC-A/W/H, side scatter-area/width/height; Tcm, central memory T cells; Teff, effector T cells; Tem, effector memory T cells; TIM-3, T cell immunoglobulin and mucin domain-containing protein 3; Tn, naïve T cells; Tregs, regulatory T cells; Trm, tissue-resident memory T cells.

2.2.2.2 FACS sort of TILs, tumor cells and fibroblasts

In cases where a sufficient number of single cells were available from the digested tumor sample (after cells were set aside for flow cytometry analysis), a minimum of 5 Mio and up to 20 Mio cells were taken for sorting of CD4$^+$ T cells, CD8$^+$ T cells, CD33$^+$ myeloid cells, tumor cells and fibroblasts. Cells were blocked for 20 min on ice in the dark using 200 – 500 µl HS, depending on the cell number. After washing with FACS buffer, 2 µl per 1 Mio cells of the respective antibodies (see Table 19) and 7-amino-actinomycin D (7AAD) for live-dead staining were added to the cell suspension in 100 – 200 µl FACS Buffer and incubated for 30 min on ice in the dark. Following a wash, cells were resuspended in 1 ml per 10 Mio cells FACS buffer, filtered and immediately used for sorting on a FACS Aria III. Single stains of each antibody were prepared using anti-IgG micro beads in accordance with the manufacturer's instructions. These stains, along with unstained cells and 7AAD-only stained cells were utilized for on-device compensation using the FACS DIVA software. Alive (7AAD negative) single cells (SSC-H low SSC-W low; FSC-H low FSC-W low) were pre-gated and the respective cell populations were sub-gated according to their surface marker expression as shown in Figure 5. The sorted cells were collected in pre-cooled tubes filled with RPMI. Subsequently, the sorted cells were pelleted, resuspended in 300 µl RNA Protect, snap-frozen, and stored in liquid nitrogen (-196 °C) for subsequent mRNA sequencing analysis. All steps were conducted on ice and as fast as possible to minimize changes in marker expression and cell viability.

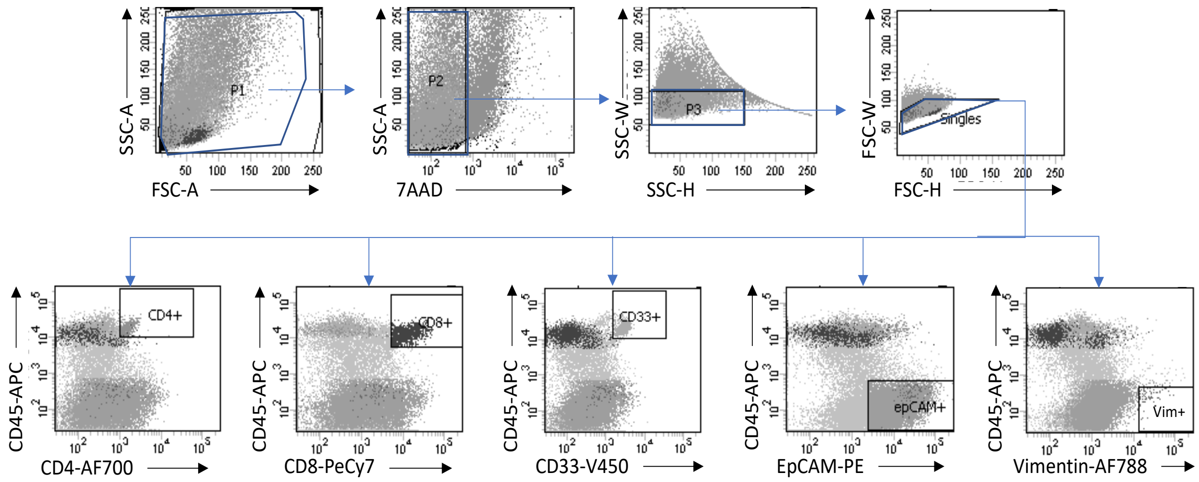

Figure 5: Gating strategy for sorting tumor-infiltrating immune cells, tumor cells and fibroblasts from fresh human tumor tissue via flow cytometry.
First all cells of all sizes were included, then all alive cells (7AAD-) were gated and single cells were selected in the following two gates. Of these cells, CD4$^+$ T cells (CD45$^+$CD4$^+$), CD8$^+$ T cells (CD45$^+$CD8$^+$) and myeloid cells (CD45$^+$CD33$^+$) were determined and gates were set for sorting of selected cells. If applicable, tumor cells (CD45-EpCAM$^+$ for carcinoma, CD45-EpCAM- for sarcoma and melanoma) and fibroblasts (CD45$^-$Vimentin$^+$) were sorted as well. Exemplary plots are shown from sorting of ImmuNEO-15. 7AAD, 7-Aminoactinomycin D; CD, cluster of differentiation; EpCAM, epithelial cell adhesion molecule; FCS-A/W/H, forward scatter-area/width/height; SSC-A/W/H, side scatter-area/width/height.

2.2.2.3 Bulk mRNA sequencing of sorted cells and analysis

RNA isolation, library preparation and mRNA sequencing were performed by AG Klink at the University of Dresden as part of the joint funding project. In brief, libraries were generated from each bulk of sorted cells using the with SMART-Seq Stranded Kit (Takara) and paired-end sequencing (2 x 75 base pairs (bp)) on a NextSeq 500 (Illumina) was performed to reach at least 50 Mio raw reads per sample.

Data processing was then performed by AG Rad at the Technical University Munich and the analysis strategy was developed and conducted in cooperation with AG Rad as described in Tretter *et al.*, 2023, methods. Prior to analysis, the quality of each library was investigated, and some patients and samples had to be excluded due to low data quality (ImmuNEO-04, -17.2 and -28). For analysis, only one representative sample per patient was used and patients were grouped into a long survival (above 1 year) and short survival (below 1 year) group according to their survival time since tumor resection/MASTER inclusion. According to these groups the gene expression profiles of the sorted cell types, focusing on CD8$^+$ T cells, were compared as described in Tretter *et al.* and gene-set enrichment analysis (GSEA) was used for pathway analyses of these genes.

2.2.3 Whole exome, whole genome and RNA sequencing of patient material and analysis

2.2.3.1 DNA and RNA sequencing

Whole exome (WES), whole genome (WGS) and RNA sequencing (RNA-seq) was performed as part by the DKFZ in Heidelberg as described in Tretter *et al.* for patients included in MASTER study but also for patients included in the ImmuNEO plus cohort only. The respective analyses performed per patient and tumor sample are listed in section 6.1.

2.2.3.2 Variant calling from DNA and RNA sequencing data

Variant calling was performed on WES/WGS and RNA-Seq data for identification of single nucleotide changes and insertion/deletions for patients with the available data sets as described in Tretter *et al.*. Furthermore, the tumor mutational burden per sample was calculated. The analysis of the complete variant data set of each patient tumor was performed with a custom R script as shown in Appendix 6.9.1.

2.2.3.3 HLA typing

The identification of each HLA class I type was performed by AG Rad at the Technical University Munich from the available WES/WGS data using xHLA (Xie et al., 2017), BWAKit (Li, 2013) and OptiType (Szolek et al., 2014) using default settings for all patients. The consensus of all three algorithms was then used for HLA typing. For confirmation of predications, additional HLA typing was performed using genomic DNA isolated from patients' PBMC through targeted next-generation sequencing. This analysis was conducted in selected patients by the Zentrum für Humangenetik und Laboratoriumsdiagnostik in Martinsried, Germany.

2.2.4 HLA class I immunopeptidome analysis

2.2.4.1 Immunoprecipitation of HLA complexes and MS analysis of eluted peptides

Immunoprecipitation of HLA class I complexes, subsequent elution and purification of bound peptide ligands as well as measurement by liquid chromatography (LC)-MS/MS analysis was performed on indicated tumor samples (see table in section 6.1) as previously described in Bassani-Sternberg *et al.* and Tretter *et al.* by Matteo Pecoraro of AG Mann at the Max-Plank Institute of Biochemistry in Munich.

2.2.4.2 Peptide identification from MS immunopeptidome data

To identify peptide sequences from the MS spectra generated in 2.2.4.1, pFIND 3.1.5 (Chi et al., 2018) was used to match all possible human protein sequences included in the reference protein database (obtained from Human Ensembl GRCh38, release 92) to the spectra files of each measured peptide. A number of known contaminates were also included (provided within pFIND) to filter sequences from potential contaminating substances. The acceptable precursor tolerance and fragment tolerance were configured at 20 parts per million (p.p.m.). Parameters were further set to search for non-specifically digested peptides with a maximum mass of 1,500 Dalton (Da), ranging from 8 to 15mers. Additionally, N-terminal acetylation (42.010565 Da), methionine oxidation (15.994915 Da) and cysteine carbamidomethylation (57.021463 Da) were considered as potential post-translational modifications. Peptides matching a protein sequence (excluding contaminants) were further utilized and filtered with a false discovery rate (FDR) of 0.01 at the peptide spectrum match level generating the immunopeptidome data set. The analysis of the whole immunopeptidome was performed using a custom R script as shown in Appendix 6.9.2.

2.2.4.3 MHC-motif deconvolution

To assess the quality and purity of the MS-generated immunopeptidomics data, MHCMotifDecon-1.0 (DTU Health Tech, 2022; Kaabinejadian et al., 2022) was used to deconvolute the identified peptide sequences by their binding motif to the respective patients HLA-allele. The algorithm employed in this analysis utilizes HLA binding predictions from NetMHCpan-4.1 (for HLA class I) to deconvolute and assign probable HLA restriction elements to the immunopeptidome data. For the analysis, all identified peptide sequences with lengths ranging from 8 to 15 amino acids and all HLA-A, B, and C alleles of each patient were utilized, applying standard settings as suggested on the website.

2.2.5 Pipeline for the identification of patient-specific neoantigen candidates from MS data

The previously developed MS-based pipeline (Bassani-Sternberg et al., 2016) for the identification of neoantigen candidates was further improved within this project and was applied to this diverse pan-cancer patient cohort as well as previously described patient Mel15 (Bassani-Sternberg et al., 2016) as described in Tretter *et al.*.

Novel features to the pipeline were integrated as follows: (1) On the genetic level, variant detection from RNA sequencing data was added to the pipeline using Strelka2 (Lange et al., 2020) by Sebastian Lange. This aimed at increasing the search space for potential neoantigen candidates to a diverse range of genetic regions (coding region, pseudogenes, introns, non-coding regions etc.) and aberrations (splice site variants, intron-inclusions, non-coding variants, etc.). Therefore, a sophisticated algorithm for generating and translating all three open reading frames (ORFs) surrounding a variant was developed and implemented by Niklas de Andrade Krätzig (VCFtranslate). (2) On the proteomic level, MaxQuant was exchanged into pFIND as a peptide calling tool (Chi et al., 2018) as it is designed to deal with ultra-large search spaces and thus makes the analysis of big data much faster. Furthermore, the machine learning tool Prosit (Gessulat et al., 2019; Wilhelm et al., 2021) was included into the pipeline, expanding the potential for correct neoantigen identification. (3) Additionally, a comprehensive post-processing filtering procedure was established together with Niklas de Andrade Krätzig and Philipp Seifert, specifically focusing on the exclusion of possible canonical peptides and single-nucleotide polymorphisms (SNPs). All analysis steps are described in detail in the following chapters.

2.2.5.1 Generation of patient-specific databases for MS-based neoantigen identification

The generation of the custom data base was developed, established and performed by Niklas de Andrade Krätzig as described in Tretter *et al.*.

The main goal was to obtain mutated peptide sequences based on each patient’s tumor mutational landscape as a reference data set to be used for the immunopeptidome analysis. Therefore, all variants called from WES/WGS and RNA-seq were incorporated into the wt DNA sequences of the most abundant transcript per gene obtained from biomart (v92), followed by translation into peptide sequences using VCFtranslate. All genes regardless of the transcript biotype were included for analysis. For non-protein-coding transcripts, artificial ORFs were determined by identifying paired start and stop-codons in all three reading frames and the respective variant was added. The same procedure was applied to protein-coding transcripts in case of start/stop-loss/gain and frameshift mutations. Additionally, for these variants, the coding sequence (CDS) was elongated into the respective untranslated region (UTR), if applicable. In cases where variants impacted splice donor or acceptor sites, the affected intron was incorporated into the CDS and newly assessed for valid ORFs. Generally, only variants resulting in amino acid changes on protein level and within valid ORFs were considered. For each affected transcript, up to three ORFs enclosing the variant site were translated into the corresponding mutated amino acid sequence. These peptide sequences, in conjunction with the immunopeptidomics data obtained from mass spectrometry, were then used for further analysis.

2.2.5.2 Identification of mutated peptides sequences from MS data

To identify mutated or aberrant HLA class I peptides, the reference wt protein database (Human Ensembl GRCh38, release 92) was combined with each patient-specific customized database containing the mutated amino acid sequences from 2.2.5.1. The peptide database search algorithm pFIND 3.1.5 (Chi et al., 2018) was then used to match the in-silico generated mutated amino acid sequences to the measured peptide spectra. The acceptable precursor tolerance and fragment tolerance were configured at 20 parts per million (p.p.m.). Parameters were further set to search for non-specifically digested peptides with a maximum mass of 1,500 Dalton (Da), ranging from 8 to 15mers. Additionally, N-terminal acetylation (42.010565 Da), methionine oxidation (15.994915 Da) and cysteine carbamidomethylation (57.021463 Da) were considered as potential post-translational modifications. The FDR for identification was set 5% at the peptide spectrum match level to increase the potential for neoantigen identification. As neoantigen candidates will further be assessed by immunogenicity assessment (2.2.7) and in-depth validation (2.2.8), this higher FDR was seen as applicable. Following the annotation of proteins to each matched peptide-ORF pair, the unfiltered peptide lists generated by pFIND were subjected to filtering at a FDR of 5% for direct utilization in subsequent post-processing (pFIND peptides). Additionally, the unfiltered pFIND lists were employed for further re-scoring and analysis using the Prosit pipeline developed by Mathias Wilhelm and Daniel Zolg (Gessulat et al., 2019; Wilhelm et al., 2021) (Prosit peptide). The rescoring method is described in detail in Wilhelm *et al.* (2021). All Prosit peptides exceeding an FDR threshold of 5% were subsequently considered for analysis.

Peptides identified through both approaches were combined and employed for subsequent post-processing.

2.2.5.3 Post-processing, filtering and MHC binding prediction of neoantigen candidates

First, the combined pFIND and Prosit output was filtered for general characteristics: Peptides matching to potential contaminants (provided within pFIND) or to reverse sequences (no biological function, used to determine statistical cutoffs by pFIND), were excluded. Additionally, the results underwent filtering to isolate sequences exclusively identified in the custom patient-specific variant databases and thus not found in the Ensembl wt database.

Second, the prefiltered list of mutated MS-peptides (neoantigen candidates) for each tumor was then assessed for multiple more specific parameters to reduce false positives: Peptides that incorporated a variant (SNVs, In/Dels, multiple substitutions) within their sequence were directly considered valid. Peptides lacking variants directly within the sequence but having variants elsewhere in their ORF underwent further assessment. Exonic SNVs outside of peptide sequences were directly excluded, while splice site variants and frameshift variants upstream of a potential neoantigen were manually checked using BLAT (Kent, 2002). Peptides were categorized as “mutated” or “non-wt” if a peptide within a non-canonical frame or a retained intron was detected. The filtered neoantigen candidates underwent additional validation through an automated protein BLAST (Altschul et al., 1990) search, conducted by Philipp Seifert. This analysis aimed at identifying potential similar wt sequences in the proteome. Neoantigen candidates with more than two matches in the canonical human protein database were eliminated, and peptides with 1-2 matches were manually verified through literature research and removed if needed. Furthermore, three distinct peptide databases - PeptideAtlas (Desiere et al., 2006), PepBank (Duchrow et al., 2009) and IEDB (Vita et al., 2019) - were utilized to filter for previously identified (immunogenic) epitopes.

After this filtering process, two different algorithms, NetMHC 4.0 (Andreatta & Nielsen, 2016) and MHCflurry 1.6.0 (models class1) (O’Donnell et al., 2018), were utilized to predict the binding affinity of each neoantigen candidate to the corresponding patient HLA alleles. The most favorable binding allele, based on predicted affinity or percentile rank, was identified for each algorithm and each peptide in collaboration with Philipp Seifert. Further analysis and plotting of the MS-based neoantigen candidate data was performed using a custom R script as shown in Appendix 6.9.4.

2.2.6 *In silico* prediction of nonameric peptide ligands

Putative mutated nonameric (9mer) peptide ligands originating from SNVs, identified through the described mutation calling in section 2.2.3.2, were predicted using an *in silico* prediction pipeline developed together with Niklas de Andrae Krätzig. First, all sequences bearing a SNV within a valid ORF were translated to 17-residue-long amino acid (aa) sequences (mutated aa in middle/9th position). Mutations less than 8 aa from the start or end of the ORF were excluded, as here no 17mer could be generated. NetMHC 4.0 (Andreatta & Nielsen, 2016) was then used to generate all possible 9mer peptides from the 17mer sequences and together with the HLA-A, HLA-B and HLA-C alleles of each patient, the MHC class I binding affinity was predicted for each HLA molecule. Nonameric peptides were then filtered by only including mutated peptides ranked as strong and weak binder (percentile rank < 2) with predicted affinity < 200 nM by NetMHC 4.0.

Data analysis and visualization was then performed using a custom R script as shown in Appendix 6.9.3.

2.2.7 In-vitro stimulation for immunogenicity assessment of identified neoantigen candidates

2.2.7.1 Accelerated cocultured DC culture using patient PBMCs

To evaluate the immunogenicity of the neoantigen candidates identified via proteogenomics, T cell responses against 79 neoantigen candidates from 21 patients were assessed in *in vitro* stimulation assays using patient derived autologous PBMCs as described in Tretter *et al.*. Therefore, the previously described accelerated cocultured DC (acDC) method was used as previously described with several modifications and improvements (Bassani-Sternberg et al., 2016; Martinuzzi et al., 2011a) as shown in Figure 6 and described in more detail in the following. Furthermore, Johannes Untch and Florian Dreyer established a modified version of this assay using CD137$^+$-based enrichment of neoantigen-stimulated T cells on day 1, that was used for several of the immunogenicity assays and is also describe below.

PBMCs from different time points per patient, if available, were used for immunogenicity screening. After thawing of the PBMC aliquot, alive cells were counted and up to 1 Mio PBMCs per well and condition were cultivated in AIM-V supplemented with 100 ng/ml interleukin (IL) 4 and 100 ng/ml granulocyte-macrophage colony-stimulating factor (GM-CSF) in a flat-bottom 96-well plate for maturation of dendritic cells (DCs). If more cells were available, neoantigen candidates were tested in multiple replicates.

After 24 hours, 1 µM of each respective synthetic neoantigen peptide (>90% purity) was added to the culture together with 0.5 ng/ml IL-7, 50 ng/ml tumor necrosis factor (TNF)-α and 10 ng/ml IL-1β. T cells non-specifically stimulated with 0.5 ng/µl PMA and 1 ng/µl Ionomycin were used as positive control, and as negative control only TCM was added to the cells. As an assay positive control, HD04 CD8$^+$ T cells transduced with the KIF2C-specific TCR 18.2 (by Florian Dreyer) and HD04 untransduced PBMCs were mixed and stimulated with 1 µM of the mutated KIF2C peptide.

After additional 24 h of peptide stimulation, 100 µl supernatant was collected for later enzyme-linked immunosorbent assay (ELISA) analysis (see 2.2.7.6). Cells where then either used for early T cell response analysis by enzyme-linked immunospot (ELISpot) as previously described (Bassani-Sternberg et al., 2016; Bräunlein et al., 2021) or enriched for specifically activated T cells using a CD137$^+$-based magnetic isolation (Ye et al., 2014). To enrich for peptide-stimulated T cells from the bulk T cell pool, activated cells expressing CD137 were isolated using the human CD137 MicroBead Kit following manufacturer's instructions.

The enriched cells were cultured in TCM supplemented with 5 ng/ml IL-7, 5 ng/ml IL-15, 30 U/ml IL-2 and 30 ng/ml OKT-3, together with 1 Mio irradiated (30 Gy) feeder PBMC. These cells were cultured and expanded for 12 days by adding IL-7 and IL-15 twice per week and IL-2 once per week.

Non-enriched cells after early ELISpot analysis or CD137^{-} cells (flow-through cells after enrichment) were re-cultured and expanded in TCM supplemented with 5 ng/ml IL-7 and 5 ng/ml IL-15. These cells were fed twice per week without the addition of feeder cells. According to the growth behaviour of both enriched and non-enriched cells, the total growth volume and the well size was increased gradually.

2.2.7.2 Accelerated cocultured DC culture using patient TILs

TILs have additionally been used for the immunogenicity assessment of neoantigen candidates as the abundance of neoantigen-reactive T cells might be increased in comparison to the blood. The general acDC method as described in 2.2.7.1 was followed using CD137^{+} enrichment.

As less antigen-presenting cells (APCs) are present in TILs than in PBMCs, autologous LCLs were used in two experiments for neoantigen-presentation on day 0. Therefore, autologous LCLs were irradiated with 60 Gy to stop their proliferation and were added at a ratio of 1:10 to each TIL well prior to stimulation with the respective neoantigen candidates.

2.2.7.3 Assessment of recall neoantigen-specific T cell responses

After 13 days of expanding the pre-stimulated PBMCs (2.2.7.1) or TILs (2.2.7.2), recall reactivities of these T cells to the synthetic mutated peptide ligands was evaluated. This assessment was conducted by measuring specific IFN-γ release by ELISpot assay (see Figure 6). Therefore, T cells were co-cultured with peptide-pulsed antigen-presenting target cells that had matching HLA class I alleles. For each neoantigen candidate the best binding HLA class I allele was determined using two different HLA-peptide binding prediction algorithms, NetMHC 4.0 (Andreatta & Nielsen, 2016) and MHCflurry 1.6.0 (models class1) (O'Donnell et al., 2018) as described in 2.2.5.3 and target cells were chosen accordingly. If possible, LCLs derived from the same patient (see 2.2.1.3) were preferentially used as target cells as here no HLA matching is needed and errors in HLA-matching due to limits of the binding prediction algorithms can be eliminated. If no autologous LCLs were available, HLA-matched LCLs or HLA-transduced T2 or C1R cell lines were used.

Up to 1 Mio target cells per tube were pulsed for 2 h in 200 µl AIM-V with either 1 µM of the selected mutated peptide or an irrelevant control peptide prior to co-culture with the T cells. After pulsing, target cells were washed twice with TCM and the co-cultures were performed with an effecter-to-target ratio (E:T) of 2:1 using 20,000 pre-stimulated T cells and 10,000 pulsed target cells per well for LCLs (in duplicate or triplicate for the mutated and control peptide, according to available T cell numbers). When using HLA-transduced T2 or C1R cells, the E:T was increased to 1:1 using 20,000 cells each, as for these cells no EBV-specific responses are expected and transduction efficiencies for the HLA molecules were below 100% in most cases. Co-cultures were set up on an IFN-γ pre-coated

ELISpot plate as described in 2.2.7.4. As negative control only pre-stimulated T cells without target cells and as a positive control PMA/Ionomycin stimulated T cells without target cells were used.
After incubation at 37°C for 72 h (without moving the ELISpot plate), 100 µl of supernatant was removed and frozen at -20°C for later ELISA analysis (see 2.2.7.6). Cells were either used for further cultivation in a new round-bottom 96-well plate with additional fresh TCM (in cases where all expanded T cells had to be used for the co-culture) or were otherwise discarded. The ELISpot plate was then developed and analysed as described in 2.2.7.4.

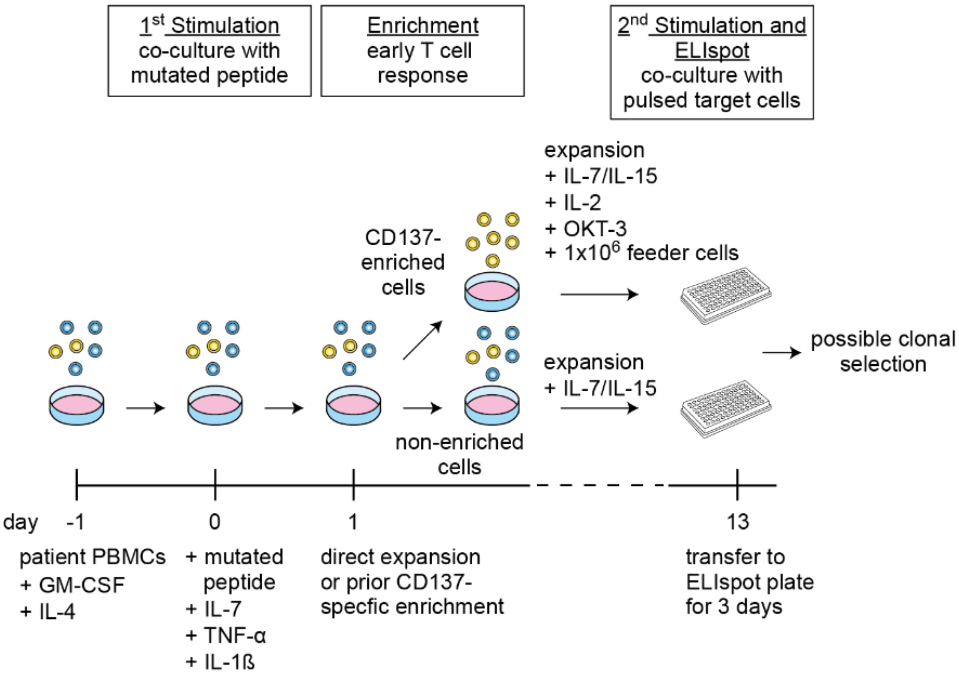

Figure 6: Schematic overview of PBMC stimulation and expansion from bulk primary PBMCs.
One day prior to peptide stimulation, up to 1 Mio. primary human PBMCs per well were taken into culture in AIM-V medium supplemented with 100 ng/ml GM-CSF and IL-4 each to stimulate DC maturation. After 24h of incubation, 1 µM synthetic peptide was added together with 0.5 ng/ml IL-7, 50 ng/ml TNF-α and 10 ng/ml IL-1β as co-stimulatory cytokines. On day 1 after peptide stimulation, cells were washed and either directly used for further expansion (lower path) or enriched for activated T cells using CD137-based magnetic enrichment (upper path). Dependent on the method, cells were expanded using different cytokines and feeder cells and after 13 days cells were re-stimulated. Therefore, HLA-matched target cells were pulsed for 2h with 1 µM of the synthetic peptide or an irrelevant control peptide and 20,000 T cells were co-cultured with 10,000-20,000 pulsed target cells for 3 days on an IFN-γ-coated ELIspot plate in duplicates or triplicates for each condition. Reactivates were assessed according to IFN-γ secretion by ELIspot analysis and positive bulk T cell populations were selected for potential T cell clone isolation. GM-CSF, granulocyte-macrophage colony-stimulating factor; HLA, human leukocyte antigen; IFN, interferon; IL, interleukin; PBMC, peripheral blood mononuclear cells; TNF, tumor necrosis factor

2.2.7.4 IFN-γ ELIspot analysis of T cell responses

96-well ELISpot plates MAHAS4510 were coated with the IFN-γ capture antibody 1-D1K (c_{END}=10µg/ml in PBS) at 4°C overnight prior to the early T cell response assessment (2.2.7.1) or recall T cell response co-culture (2.2.7.3). After the incubation with stimulated cells, ELISpot plates were washed six times using washing buffer. Subsequently, bound IFN-γ was detected by incubation with 2 µg/ml of the secondary anti-IFN-γ detection antibody 7-B6-1-biotin in 100 µl PBS + 0.5% BSA per well for two hours at room temperature (RT). After discarding the detection antibody and six additional washing steps with washing buffer, streptavidin-HRP complex solution was added to each well following a 1 h incubation at RT in the dark. Plates were subsequently washed twice with washing buffer and twice with PBS before adding 100 µl of AEC solution to each well for development of spots. The reaction was allowed to incubate in darkness until the positive control became visible, typically within a range of 2 to 15 minutes Stopping of further spot formation was then achieved by washing the plate with running tap water. Afterwards, plates were dried on towels and stored in the dark until read out.

ELISpot plates were read out using an ImmunoSpot S6 Ultra-V Analyzer with Immunospot software 5.4.0.1. Spot counts were exported and further analysed using Excel and R.

The immunogenicity of a neoantigen was defined by the spot counts at day 13, comparing the mean spots from the mutated peptide condition against the mean spots from the control peptide condition. In this study, reactivity/positive response was considered when the ratio exceeded 2, signifying that the mutated peptides induced an IFN-γ response in at least twice as many T cells compared to the control. Additionally, a difference of spots above 50, defined as the background threshold for unspecific stimulation, was used as a threshold.

Further analysis and plotting of the immunogenicity data was performed using a custom R script as shown in Appendix 6.9.5.

2.2.7.5 Clonal selection of neoantigen-specific T cells by limiting dilution

For the isolation of neoantigen-specific T-cell clones, neoantigen-reactive expanded T-cell lines (determined by ELISpot co-culture analysis in 2.2.7.1 to 2.2.7.4) were diluted to a final concentration of 0.5 to 10 cells per well and plated together with 50,000 irradiated feeder PBCMs per well in TCM supplemented with 5 ng/ml IL-7 and IL-15, 30 U/ml IL-2 and 30 ng/ml OKT-3. IL-2 was added once a week and IL-7 and IL-15 twice a week. Clones were checked for growth regularly and screened for specific reactivity once enough cells were available. Therefore, half of the cell suspension was used for a co-culture-based specificity test as described below, while the other quarter was stored in Trizol for later potential RNA isolation and the remaining cells were re-stimulated with feeder PBMCs and OKT-3. If after two weeks no cells were growing, all cells were re-stimulated with feeder PBMCs, IL-2 and OKT-3 once and trashed after further two weeks of culture.

The specificity test was performed by coculturing half of the expanded clones with 10,000 neoantigen-pulsed and control peptide-pulsed HLA-matched target cells in duplicates (same cells as used in 2.2.7.3) for 24 h at 37°C. Afterwards, 150 µl of supernatant was removed and used for analysis of IFN-γ secretion by ELISA assay.

2.2.7.6 IFN-γ ELISA analysis of T cell responses

ELISA were performed to quantify secreted cytokines like IFN-γ within the medium supernatant of T cell and target cell co-cultures from several different experiments (2.2.7.1, 2.2.7.2, 2.2.7.3 and 2.2.7.5)

The BD OptEIA™ Human IFN-γ ELISA Set from BD Bioscience was used following to the manufacturer's instructions with minor modifications. In summary, ELISA plates were coated with 50 µl IFN-γ capture antibody (1:250 diluted in coating buffer) and incubated overnight at 4°C. After three washes with washing buffer, plates were blocked with blocking solution and incubated at RT for 1 h. IFN-γ standards were freshly prepared before each experiment. Therefore, the stock solutions were dissolved in TCM or AIM-V (depending on the medium used for the coculture) to a concentration of 1000 pg/ml and five serial 1:1 dilutions as well as one blank containing only medium were prepared. Blocked plates were washed before adding 50 µl of the supernatants or standards to the well. Following incubation for 1 h at RT, plates were washed five times, and 50 µl detection solution containing Detection antibody (IFN-γ biotinylated; 1:250 diluted in blocking solution) and Enzyme Conjugate (Streptavidin-Horseradish peroxidase, 1:250 diluted in blocking solution) was added. After 1 h incubation in the dark at RT, plates were washed for seven times. Subsequently, 100 µl substrate solution (1:1 mix of A and B from BD OptEIA™ TMB Substrate Reagent Set) was added per well and plates were incubated in the dark at RT for 10-20 min. The reaction was stopped by addition of 50 µl of sulfuric acid as soon as the standard curve become completely visible. The enzymatic reaction intensity was measured with an absorbance at 450 nm and a reference of 570 nm using a Sunrise^{TM} absorbance reader.

2.2.8 Neoantigen candidate validation

2.2.8.1 Peptide verification

For peptide verification two different approaches were followed. First, 88 of the synthetic peptides ordered from DGPeptidesCo Ltd. (>90% purity) were measured by LC-MS/MS by AG Küster and Matteo Pecoraro (AG Mann) as described in detail in Tretter *et al.*. Additionally, the spectrum for each neoantigen candidate peptide was predicted by AG Küster using Prosit. Both, the synthetic and the predicted spectrum were compared to the experimental peptide spectrum from the tumor, the normalized spectral contrast angle (SA) was calculated, and the best SA was reported.

Secondly, the disparity in retention time (RT) between the mutated peptide and the retention time predicted by Prosit was compared to all measured peptides. The resulting RT errors were then reported following the approach outlined in Tretter *et al.*.

2.2.8.2 Prevalence of variants in human healthy tissue RNA-seq data

To assess the tumor-specificity of potential neoantigen candidates, the prevalence of all neoantigen candidate variants was assessed in 10,269 RNA-seq samples across 30 different human healthy tissues derived from the Genotype-Tissue Expression (GTEx) Portal (Lonsdale et al., 2013) as described in Tretter *et al.* in cooperation with Niklas de Andrade-Krätzig. A variant was considered found with at least one supporting hit and not available if the position was covered (with min 3 reads) in less than 5% of GTEx samples.

Furthermore, the RNA-seq data of sorted CD8$^+$ TIL (2.2.2.3) was investigated for the presence of all neoantigen candidate variants to search for patient specific RNA editing events.

2.2.9 Statistical analysis

A two-tailed Mann-Whitney U-test was employed to compare the frequencies of CD8$^+$ T cells expressing a minimum of one activation marker (HLA-DR, CD103) in tumor samples with high versus low immune cell infiltration.

Correlations between two independent continuous parameters were evaluated using Spearman's rank correlation coefficient. For the correlation between the numbers of DNA and RNA variants within one tumor, only samples having both data sets were utilized. Furthermore, for the correlation of the phenotypic data with the peptidomic data, only one representative tumor specimen per patient was utilized for statistical analyses (ImmuNEO core cohort see 6.1). This approach was implemented to mitigate bias arising from multiple metastases available for some patients. Custom R scripts can be found in Appendix 6.9.7.

Kaplan-Meier curves, along with log rank test and Cox's proportional hazards model, were employed to assess the overall survival since tumor resection between distinct groups of patients within the ImmuNEO cohort. For continuous parameters, groups were stratified based on the median into "high" (above median) and "low" (below median) groups. In the case of relative parameters (frequencies, 0-100%), patients were categorized into a "high" group with fractions above 50% and "low" group with fractions below 50%. Again, to mitigate bias due to multiple metastases available for some patients, only one representative tumor sample from each patient (ImmuNEO core cohort see 6.1) was used. Custom R scripts can be found in Appendix 6.9.7.

The immunophenotyping data of the TME and the size of the immunopeptidome was compared to the quantity of validated immunogenic and non-immunogenic neoantigen candidates by using two-tailed Mann-Whitney U-test with Benjamini-Hochberg correction.

Two-tailed Mann-Whitney U-test was used to compare the number of total mutations with the response of patients to ICI. Kruskal-Wallis rank sum test (H-test) for overall correlations and subsequent Two-tailed Mann-Whitney U-test for pair-wise correlations was used to correlate several experimental features to the response type of patients to ICI divided into no, mixed and good response. Custom R scripts can be found in Appendix 6.9.7.

2.2.10 Data availability

The mass spectrometry proteomics data can be found via the dataset identifier PXD037655 on the PRoteomics IDEntifications Database PRIDE (Perez-Riverol et al., 2022).

The WES/WGS and RNA-seq data of all samples have been submitted to the European Genome-phenome Archive (*EGA European Genome-Phenome Archive*, n.d.) with the study identifier EGAS00001006706. As this data set contains sensitive patient information, the data is only available upon request from the associated Data Access Committee. Via the study accession number PRJEB61429 the RNA-seq data from sorted CD8^{+} TILs can be found in the European Nucleotide Archive (*ENA European Nucleotide Archive*, n.d.).

3. Results

3.1 Patient cohort and study design

This study was part of a Joint Funding Initiative of the German consortium for translational cancer research (Deutsches Konsortium für translationale Krebsforschung, DKTK) that took advantage of a large prospective patient cohort compiled for the MASTER Program (Horak et al., 2021). Comprehensive details regarding patient samples and corresponding analyses are provided in the methods section and are summarized in Table 24 (more detailed table and further information on HLA-types and therapies see Appendix 6.1 and 6.2, and material 2.1.3). This data was already published in Tretter *et al.*. In short, within this study 51 fresh tumor samples and over 120 blood samples from in total 42 patients were prospectively collected. Of these, twelve samples from seven patients were primarily not included into the MASTER program and were added as ImmuNeo Plus samples to the cohort. 42 tumor samples from 32 patients were finally fully eligible for analysis (ImmuNEO cohort) from which one representative tumor sample was selected per patient for the following statistical analysis (ImmuNEO core cohort) (see Table 24 and Appendix 6.1 and 6.2). The ImmuNeo cohort includes 25 different tumor entities, including a diversity of sarcomas, carcinomas as well as melanomas (see Figure 7 a) at different stages and with diverse metastatic sites. Therapies applied in these patients also vary with around 40% of patients having received ICI (Appendix 6.2) and the overall survival of all ImmuNEO patients is summarized in Figure 7 b.

For the identification of neoantigen candidates in this cross-entity cohort ImmuNEO MASTER and for the identification of related common tissue-agnostic hallmarks, a comprehensive workflow for the analyses of tumor specimens was developed which is depicted in Figure 8.

Table 24: Overview of the ImmuNEO patient cohort.

Details for each tumor sample in the ImmuNEO cohort are provided, including information on the tumor entity, metastatic site (or primary site), and the primary sampling cohort. Core samples used for statistical analysis are highlighted in bold. Tumor samples utilized for immune phenotyping of the tumor immune microenvironment (IME) through flow cytometric assessment and RNA sequencing (RNA-seq) of sorted CD8^{+} T cells are labeled. Additionally, samples subjected to whole exome sequencing (WES) and bulk tumor RNA-seq are annotated, while those analyzed via whole genome sequencing (WGS) are indicated with an asterisk. The survival status, along with the survival time in months since admission to MASTER/tumor resection (MASTER), is presented. Further information includes details on patients who received immune checkpoint blockade, specifying the response as no response (0), mixed response (1), or good response (2). Ca, carcinoma; DSRCT, desmoplastic small round cell tumor; MPNST, malignant peripheral nerve sheath tumor; GIST, gastrointestinal stromal tumor; LN, lymph node; IN, ImmuNEO; MS, mass spectrometry; WES, whole exome sequencing; WGS, whole genome sequencing; RNA-seq, RNA sequencing; IME, immune microenvironment. (Tretter et al., 2023)

General information				Subset	IME analysis		MS	Mutation Analysis		Survival data [months]		Immunotherapy Checkpoint Therapy	
Patient ID:	**Tumor entity**	**Metastatic site**	**Cohort**	**Core Samples**	**Pheno-typing**	**Sort & RNAseq**	**MS**	**WES (*WGS)**	**RNAseq tumor**	**Survival status**	**since MASTER**	**Received general**	**Response**
ImmuNEO-1.1	Thymoma	lung	MASTER	✓	x	x	✓	✓	✓	alive	46	Yes	2
ImmuNEO-1.2		lung pericardium	-	x	x	x	x	x	x				
ImmuNEO-2	Mamma-Ca	primary	MASTER	✓	x	x	✓	✓	✓	alive	37	x	-
ImmuNEO-3	Sarcoma (DSRCT)	primary	MASTER	✓	x	x	✓	✓	✓	deceased	23	x	-
ImmuNEO-4	Renal-cell-Ca	LN	MASTER	✓	✓	x	✓	✓	✓	deceased	12	Yes	2
ImmuNEO-5	Leiomyosarcoma	lung	MASTER	✓	✓	x	✓	✓	✓	deceased	4	x	-
ImmuNEO-8	Ovarian-Ca	hypogastrium	MASTER	✓	✓	x	✓	✓	✓	deceased	21	Yes	0
ImmuNEO-9	Thyroid-Ca	LN	MASTER	✓	x	x	✓	✓	✓	alive	39	x	-
ImmuNEO-11.1	Endometrium-Ca	primary	MASTER	x	✓	✓	✓	✓	x	alive	32	x	-
ImmuNEO-11.2	Pancreas-Ca	LN	MASTER	✓	✓	✓	✓	✓	✓				
ImmuNEO-13	Testicle-Ca	LN	MASTER	✓	x	x	✓	✓	✓	deceased	10	x	-
ImmuNEO-14	Melanoma	abdominal wall	MASTER	✓	x	x	✓	✓	x	deceased	6	Yes	0
ImmuNEO-15	Testicle-Ca	lung	MASTER	✓	✓	✓	✓	✓	✓	alive	35	Yes	0
ImmuNEO-16	Adeno-Ca	primary	MASTER	✓	x	x	✓	✓	x	deceased	17	Yes	0
ImmuNEO-17.1	Melanoma	LN	IN Plus	x	x	x	✓	x	x	alive	31	Yes	1
ImmuNEO-17.2		LN	IN Plus	✓	✓	x	✓	✓	✓				
ImmuNEO-17.3		LN	IN Plus	x	x	x	✓	✓	✓				
ImmuNEO-18	Mamma-Ca	ovar	IN Plus	✓	✓	x	✓	✓	✓	deceased	22	x	-
ImmuNEO-19.1	Melanoma	LN colon	IN Plus	x	✓	✓	✓	✓	✓	alive	29	Yes	2
ImmuNEO-19.2		colon	IN Plus	x	✓	✓	✓	✓	✓				
ImmuNEO-19.3		colon	IN Plus	x	✓	✓	✓	x	x				
ImmuNEO-19.4		liver	IN Plus	✓	✓	✓	✓	✓	✓				
ImmuNEO-20	Testicle-Ca	LN	MASTER	✓	✓	✓	✓	✓	x	deceased	8	x	-
ImmuNEO-22	Melanoma	abdominal wall	MASTER	✓	x	x	✓	✓*	✓	alive	31	Yes	1
ImmuNEO-23.1	Sarcoma (MPNST)	LN	IN Plus	x	✓	x	✓	✓	✓	deceased	8	x	-
ImmuNEO-23.2		primary (thorax)	MASTER	✓	✓	✓	✓	✓	✓				
ImmuNEO-24.1	Adrenocortical-Ca	liver	IN Plus	✓	✓	✓	✓	✓	✓	alive	32	x	-
ImmuNEO-24.2		primary (kidney)	MASTER	x	✓	✓	✓	✓	✓				
ImmuNEO-25	Sarcoma (GIST)	primary (intestine)	MASTER	✓	✓	✓	✓	✓	x	alive	15	x	-
ImmuNEO-26	Adeno-Ca	primary	MASTER	✓	x	x	✓	✓	✓	deceased	5	Yes	1
ImmuNEO-27.1	Fibrosarcoma	primary	IN Plus	x	x	x	✓	✓	✓	alive	26	x	-
ImmuNEO-27.2		lung	MASTER	✓	x	x	✓	✓	✓				
ImmuNEO-28	Clear cell sarcoma	primary	MASTER	✓	✓	x	✓	✓	✓	alive	25	x	-
ImmuNEO-30	Synovial sarcoma	primary	MASTER	✓	x	x	✓	✓*	✓	alive	26	x	-
ImmuNEO-31	Rhabdomyosarcoma	primary	MASTER	✓	x	x	✓	✓	x	deceased	14	x	-
ImmuNEO-32	Osteosarcoma	brain	MASTER	✓	✓	✓	✓	✓	✓	deceased	2	x	-
ImmuNEO-33	atypical carcinoid of the lung	asubcut. thorax	MASTER	✓	x	x	✓	✓	✓	deceased	5	x	-
ImmuNEO-34	Adeno-Ca	primary	MASTER	✓	x	x	✓	✓	x	deceased	3	x	-
ImmuNEO-35	Fibrosarcoma	*n.a.*	MASTER	✓	✓	x	✓	✓	✓	alive	9	x	-
ImmuNEO-36	Adeno-Ca (Barret-Ca)	LN	MASTER	✓	x	x	✓	✓	✓	deceased	6	Yes	0
ImmuNEO-37	Adeno-Ca	primary	MASTER	✓	✓	x	✓	✓	✓	deceased	4	x	-
ImmuNEO-38	Sarcoma (MPNST)	colon	MASTER	✓	✓	x	✓	✓	✓	alive	13	x	-

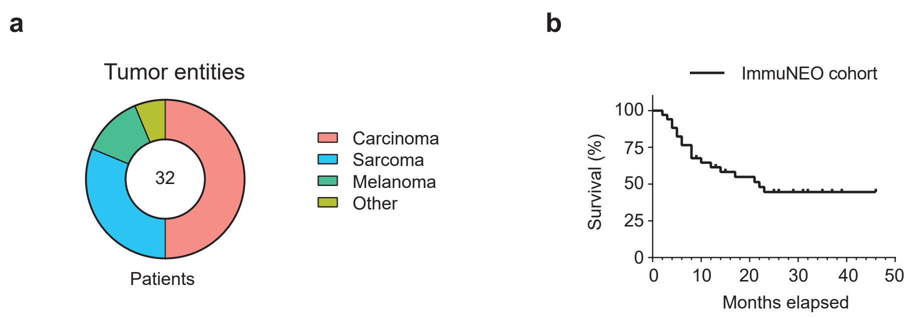

Figure 7: ImmuNEO MASTER cohort.

a, The graph illustrates the distribution of the primary tumor entities among patients in the ImmuNEO MASTER cohort. **b,** Displayed is the overall survival of patients in the ImmuNEO MASTER cohort measured in months since tumor resection. **a,b,** n = 32 patients (see Table 24 and 6.1). (Tretter et al., 2023)

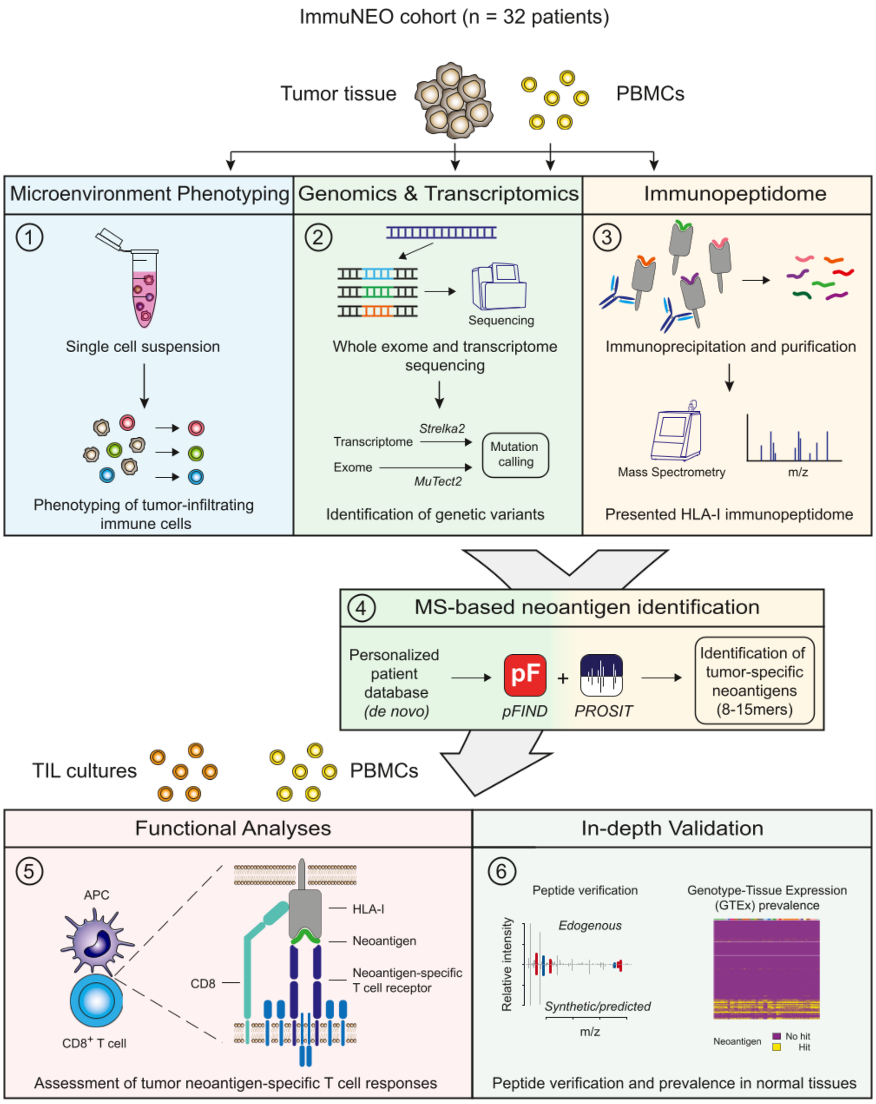
ImmuNEO cohort (n = 32 patients)
Tumor tissue
PBMCs
Microenvironment Phenotyping
Genomics & Transcriptomics
Immunopeptidome
1
Single cell suspension
Phenotyping of tumor-infiltrating immune cells
2
Sequencing
Whole exome and transcriptome sequencing
Strelka2
Transcriptome
Exome
MuTect2
Mutation calling
Identification of genetic variants
3
Immunoprecipitation and purification
Mass Spectrometry
m/z
Presented HLA-I immunopeptidome
4
MS-based neoantigen identification
Personalized patient database (de novo)
pF
pFIND
+
PROSIT
Identification of tumor-specific neoantigens (8-15mers)
TIL cultures
PBMCs
Functional Analyses
In-depth Validation
5
APC
CD8+ T cell
CD8
HLA-I
Neoantigen
Neoantigen-specific T cell receptor
Assessment of tumor neoantigen-specific T cell responses
6
Peptide verification
Relative intensity
Edogenous
Synthetic/predicted
m/z
Genotype-Tissue Expression (GTEx) prevalence
Neoantigen
No hit
Hit
Peptide verification and prevalence in normal tissues

Figure 8: Overview of the workflow for immunophenotyping, proteogenomic and functional analyses for neoantigen identification in the cross-entity cohort.

In this study, tumor material and peripheral blood from 32 patients included into the ImmoNEO MASTER cohort was sampled and analysed using the subsequent methods: **(1)** tumor microenvironment phenotyping; fresh primary tumor tissue underwent enzymatic digestion, and single cells were analysed using multi-color flow cytometry for the detection of multiple immune cells and phenotypic markers. Additionally, CD8$^+$ T cells were FACS-sorted and used for bulk transcriptome analysis through RNA sequencing (RNA-seq). **(2)** Genomic and transcriptomic analysis; whole exome sequencing (WES), whole genome sequencing (WGS) and RNA-seq was performed on primary tumor tissue. Blood samples from the same patient served as normal controls for WES/WGS analyses. Mutations were identified using MuTect2 v4.1.0.0 (WES/WGS) and Strelka2 v2.9.10 (RNA-seq), including filtering for single nucleotide polymorphisms (SNPs) using dbSNP. **(3)** Immunopeptidome analysis; HLA class I-bound peptide immunoprecipitation was performed from fresh primary tumor tissue and eluted peptides were sequenced using mass spectrometry (MS) analysis. The entire HLA class I peptidome (8-15mers) was subsequently analysed using pFIND (v3.1.5) with 1% FDR. **(4)** MS-based neoantigen identification; patient-specific variant data from genomic and transcriptomic analyses (2) was utilized to create a personalized variant-peptide database for each patient using VCF-translate v1.5. MS-identified peptide sequences (3) were matched to this personalized database using pFIND with 5% FDR and the machine learning tool Prosit. Filtering and post-processing steps were then applied for the identification of tumor-presented neoantigen candidates. **(5)** Immunogenicity assessment of neoantigen candidates; patient-derived autologous immune cells (PBMCs and TILs) and selected allogenic-matched healthy donor-derived PBMCs were used for immunogenicity testing of the identified neoantigen candidates. Therefore, a modified accelerated co-cultured dendritic cell (acDC) protocol was employed. **(6)** In-depth validation of peptides and variants; additional validation steps of peptides and variants of the neoantigen candidates were performed. Peptide verification involved comparing the MS-identified peptides with synthetic peptide spectra and predicted spectra as well as the respective retention times using Prosit. Furthermore, RNA variants were validated for tumor specificity using healthy tissue RNA-seq data from the GTEx database. Neoantigen candidates were prioritized based on these validation criteria. APC, antigen-presenting cell; FDR, false discovery rate; HLA-I, human leukocyte antigen class I; ORF, open reading frame; m/z, mass/charge number of ions; PBMC, peripheral blood mononuclear cells; TIL, tumor-infiltrating lymphocytes. (Tretter et al., 2023)

First (Figure 8 – 1), an assessment of tumor-infiltrating immune cells within the tumor microenvironment (TME) of fresh tumor tissue was conducted This involved employing flow cytometric immunophenotyping (2.2.2.1) as well as transcriptome analyses by bulk RNA-seq of sorted CD8^{+} T cells (2.2.2.2 and 2.2.2.3).

Next (Figure 8 – 2), for the mutational characterization of tumor samples, WES/WGS and RNA-seq data from patients' tumors measured at the German Cancer Research Centre (Deutsches Krebsforschungzentrum, DKFZ) core facility in Heidelberg (2.2.3) were used, and mutations were called from the DNA and RNA sequencing data using Mutect2 and StrelkaRNA. Subsequently, an analytical script for the analysis of the total genomic and transcriptomic data set aiming at discovery of shared mutational patterns was developed within this work (see Appendix 6.9.1).

At the heart of the neoantigen discovery pipeline lies its proteogenomic approach.

This involved immunoprecipitation of pHLA-I complexes followed by MS analysis of eluted peptides to measure the total presented immunopeptidome (Figure 8 – 3). For the analysis and characterization of the individual immunopeptidomes and potential common features between patients, pFIND was established at our group and an R-based analysis script for in-depth analysis of the immunopeptidomic data set was created (see Appendix 6.9.2).

We then developed and employed an enhanced workflow of our previously published strategy (Bassani-Sternberg et al., 2016) for neoantigens identification. This proteogenomics approach combined the personalized genomic data with the MS-based immunopeptidomic data using pFIND (Chi et al., 2018)(Figure 8 – 4). As critical innovations, variants called from RNA-seq data were included and the artificial intelligence algorithm Prosit was implemented for re-scoring of the peptide-spectra matching to increase coverage and sensitivity of our neoantigen discovery pipeline (Gessulat et al., 2019; Wilhelm et al., 2021).

To characterize the identified MS-based neoantigen candidates, the immunogenicity of the respective neoantigen candidates was assessed *in vitro* (Figure 8 – 5) by using patient-derived autologous PBMCs as well as TILS in optimized acDC assays. Also, immunogenicity tests with some identified neoantigen candidates using healthy donor (HD)-derived allogenic-matched T cells were performed.

Furthermore, an in-depth validation of neoantigen candidates using peptide verification and tumor specificity assessment was performed (Figure 8 – 6).

Ultimately, to unravel potential clinical implications for neoantigen identification, we examined the correlation between the total and immunogenic neoantigens and the immunophenotyping data of the TME. Furthermore, the influence of several biomarkers identified within this multi-omics data on patient survival and response to ICI was assessed.

3.2 Phenotyping of the tumor microenvironment

3.2.1 Flow-cytometry based phenotyping of several immune cell subtypes

To characterize the obtained primary tumor samples at a cellular level and investigate potential tumor-agnostic immunological features within the immune TME, flow cytometric immunophenotyping was conducted on fresh primary tumor tissues, as already published in Tretter *et al.*. Therefore, T cell subsets as well as natural killer cells (NK cells) were examined (gating strategy for all cells see Figure 4 in section 2.2.2.1, example for T cell subsets of ImmuNEO-19 T4 is shown in Figure 9 a) in 17 patients, from whom sufficient tumor material was accessible. The quantified cell numbers per gram tumor tissue were calculated and displayed in Figure 9 b showing a group of samples with a generally high infiltration of immune cells.

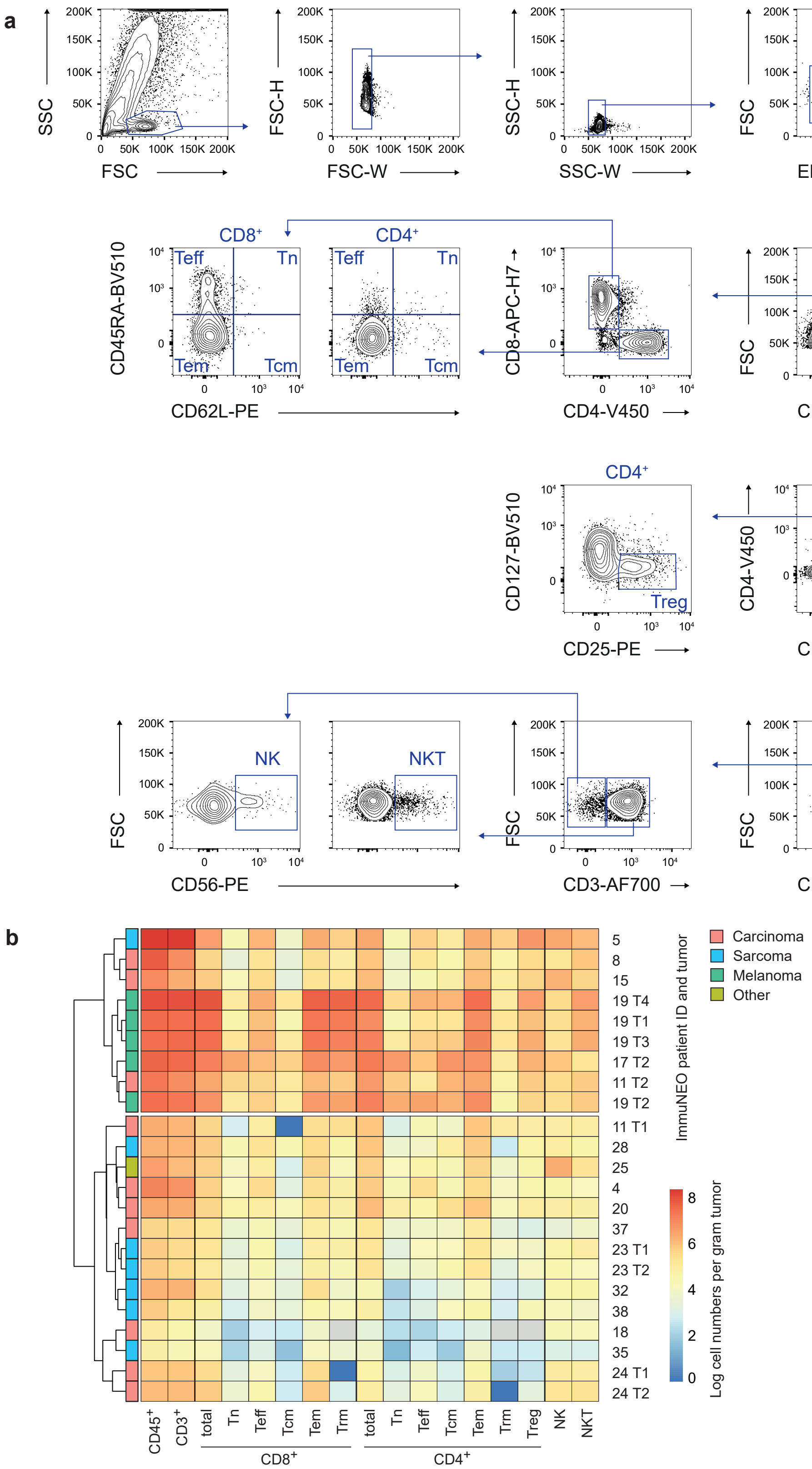

Figure 9: Tumor microenvironment analysis.

a, Gating strategy for flow cytometric analysis of $CD4^+$ and $CD8^+$ T cell subsets. **b**, Heatmap illustrating log10-transformed quantified cell numbers per gram of tumor for distinct immune cell subpopulation identified through flow-cytometry analysis of fresh tumor tissue. Patient samples underwent hierarchical clustering, categorizing them into groups with high and low immune cell infiltration. Missing values are indicated in grey. **b,** n = 23 tumor samples from n = 17 patients (see Table 24). CD, Cluster of differentiation; EMA, ethidium monoazide; FCS-A/W/H, forward scatter-area/width/height; NK, natural killer; NKT, natural killer T cells; SSC-A/W/H, side scatter-area/width/height; T, tumor; Tcm, central memory T cells; Teff, effector T cells; Tem, effector memory T cells; Tn, naïve T cells; Tregs, regulatory T cells; Trm, tissue-resident memory T cells. (Tretter et al., 2023)

For the further detailed analysis of immune cell subsets, we focused on CD8^{+} and CD4^{+} T cells only, as these cells are mainly involved in neoantigen-directed immunity due to their ability to bind HLA class I and II molecules.

First, the relative cell numbers of CD8^{+} T cells per gram tumor were assessed in more detail (Figure 10 a). The two melanoma patients with several metastasis and the pancreatic cancer metastasis of a patient with dMMR (ImmuNEO-11 T2) showed high levels of T-cell infiltration matching to the high TMB often described for these malignancies (Alexandrov et al., 2013; Le et al., 2017). However, also other tumor entities, including a sarcoma sample (ImmuNEO-5), demonstrated high numbers of TILs in general and CD8^{+} T cells specifically (Figure 9 b and Figure 10 a).

Second, we had a closer look at the distribution of T cell subsets. Effector memory T cells (Tem; CD45RA^{-}CD62L^{low}) were the prominent subset observed in both, CD8^{+} and CD4^{+} T cells, regardless of the specific tumor entity (Figure 10 b and c). Moreover, the distribution of CD8^{+} T cell subsets showed high similarity across various metastases within an individual patient, and this pattern remained consistent regardless of their anatomical metastatic location (Figure 10 b), despite variations in their relative cell numbers (Figure 10 a). This was observed to a lesser extend also for CD4^{+} T cells (Figure 10 c).

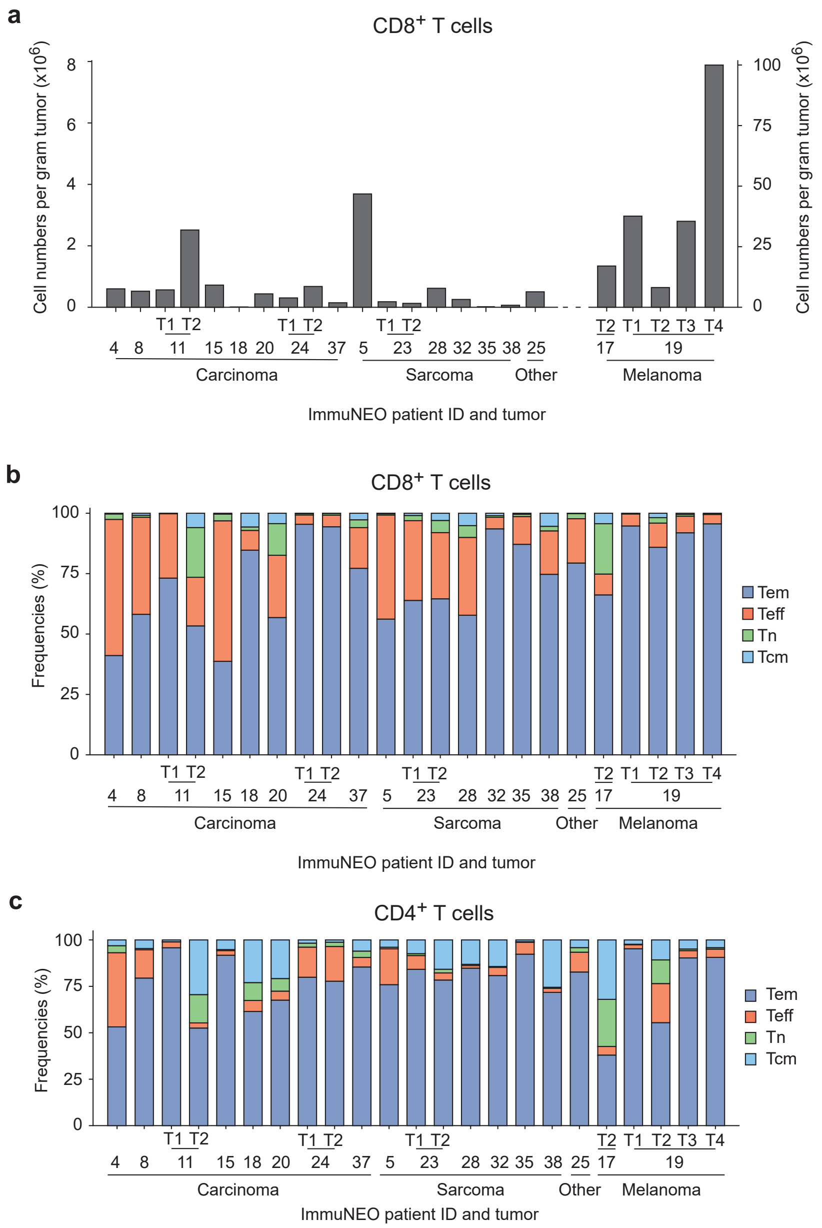

Figure 10: Phenotypic analysis of CD8⁺ and CD4⁺ T cells and their subtypes in the ImmuNEO MASTER cohort.

a, Quantitative counts of CD8⁺ T cells per gram of tumor identified through flow cytometric assessment of fresh tumor tissue per patient, categorized by tumor entity. **b, c** Frequencies of distinct CD8⁺ T cell **(b)** and CD4⁺ T cell subsets **(c)** among all identified tumor-infiltrating CD8⁺/CD4⁺ T cells per patient, sorted by tumor entity. **a-c,** n = 23 tumor samples from n = 17 patients (see Table 24). T, tumor; Tcm, central memory T cells; Teff, effector T cells; Tem, effector memory T cells; Tn, naïve T cells. (Tretter et al., 2023)

Furthermore, the functional state of the tumor-infiltrating CD8$^+$ and CD4$^+$ T cells, and thus their potential anti-tumor activity, was characterized by analyzing the surface expression of selected activation markers (HLA-DR and CD103) and inhibitory markers (PD-1, TIM-3, and LAG-3). To address differences in overall cell numbers and to explore the activation status at a population level, the frequencies of CD8$^+$ and CD4$^+$ T cells (Figure 11 a) expressing at least one activation or inhibitory marker were assessed. No statistically significant differences were noted in the frequencies of CD8$^+$ T cells with activation or inhibitory markers across different tumor entities (Figure 11 b). Thus, tumor specimens with elevated frequencies of inhibitory markers can be found in carcinoma, sarcoma, and melanoma patients, without distinct patterns in this patient population (Figure 11).

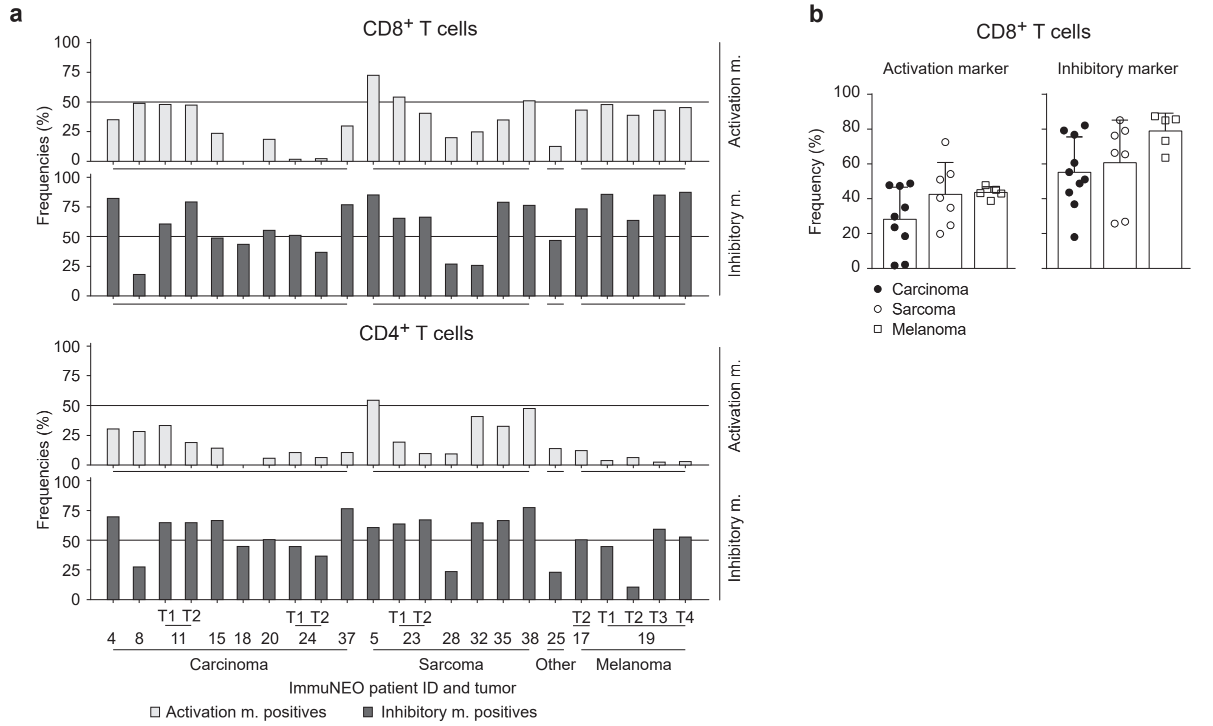

Figure 11: Analysis of the functional state of CD8⁺ and CD4⁺ T cells in the tumor microenvironment.
a, Representation of CD4⁺ (bottom) and CD8⁺ T cells frequencies (top) per patient expressing at least one activation marker (HLA-DR, CD103) or inhibitory marker (PD-1, TIM-3, LAG-3). **b,** Comparison of the proportions of CD8⁺ T cells expressing at least one activation marker (HLA-DR, CD103) or inhibitory marker (PD-1, TIM-3, LAG-3) between the mayor cancer entities. Symbols represent individual tumor samples and data are presented as mean + standard deviation. **a,** n = 23 tumor samples from n = 17 patients (see Table 24). **b,** n = 22 tumor samples of n = 16 patients. m., marker; T, tumor. (Tretter et al., 2023)

To uncover tissue-agnostic features that correlate with survival, the impact of each parameter on the patients' survival since tumor resection was evaluated using the log rank test and Cox's proportional hazards model. While elevated quantified cell numbers and overall frequencies of CD8⁺ T cells within the tumor displayed a non-significant trend towards improved survival, a higher total frequency of CD8⁺ T cells devoid of inhibitory markers correlated positively with increased survival (Figure 12 a). Similarly, the frequencies of cells within the CD8⁺ Teff subset without any marker expression, whether activation or inhibitory, also exhibited a positive correlation with increased survival. Conversely, a high fraction of cells within this subset expressing activation or inhibitory markers showed the opposite effect by positively correlating with reduced survival. Notably, only non-significant trends for CD4⁺ T cells were observed (Figure 12 b).

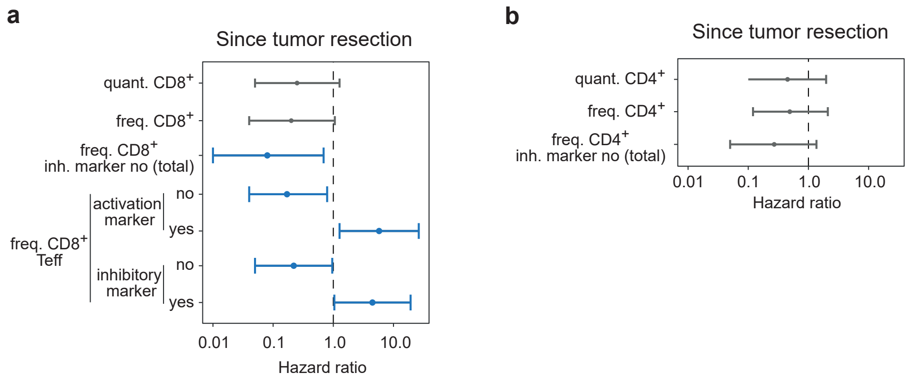

Figure 12: Correlation between phenotypic features and patient survival since tumor resection.
a, Forest plot depicting the hazard ratio (dot) and 95% confidence intervals (lines), calculated through log-rank test and Cox's proportional hazards model, of various phenotypic parameters of CD8$^+$ T cells in relation to the survival of patients since tumor resection (n = 17). **b,** Forest plot for CD4$^+$ T cells. For both, significant correlations (p ≤ 0.05) are highlighted in blue. Only one representative tumor sample per patient was used for statistical testing (see core cohort Table 24). freq., frequency; inhib., inhibitory; quant., quantified cells per gram tumor; Teff, effector T cells. (Tretter et al., 2023)

3.2.2 Gene expression patterns of sorted CD8$^+$ T cells defined by RNA-seq

In order to characterize the cells of the TME in more detail and to identify clinically relevant transcriptional signatures within this cohort, several different cell types from the TME including CD8$^+$ and CD4$^+$ T cells (CD45$^+$CD8$^+$/CD4$^+$), myeloid cells (CD45$^+$CD33$^+$) were FACS-sorted from digested primary tumor tissue (gating strategy example of ImmuNEO-15 see Figure 5 in section 2.2.2.2). This data is already published in Tretter *et al.*.

Sorted cells from 16 tumor samples and 11 patients were sent for bulk RNA-seq analysis, however only 13 samples from 8 patients were eligible for analysis (see Table 24) as the RNA quality used for sequencing was not sufficient for library establishment. To conduct the analysis, patients were categorized into two groups based on their survival data since tumor resection: a short survival group (less than 1 year) and a long survival group (more than 1 year) (Figure 13 a and Table 24/Appendix 6.1). The gene expression levels of the sorted immune cells were compared between these two groups, again focusing on CD8$^+$ T cells. Gene set enrichment analyses (GSEA) revealed that pathways linked to T cell-mediated cytotoxic functions were upregulated in the long survival group. Conversely, pathways associated with a general inflammatory response were found to be upregulated in the patient group with short survival (Figure 13 b).

In summary, tumor-infiltrating T cells within this diverse pan-cancer cohort were predominantly composed of Tem cells, irrespective of the specific tumor entity. The functional state of CD8$^+$ T cells, particularly CD8$^+$ Teff cells, demonstrated a significant impact on the overall survival of patients. Also, a more cytotoxic immune activation profile of CD8$^+$ T cells rather than a general inflammatory profile shows a beneficial influence on patient survival.

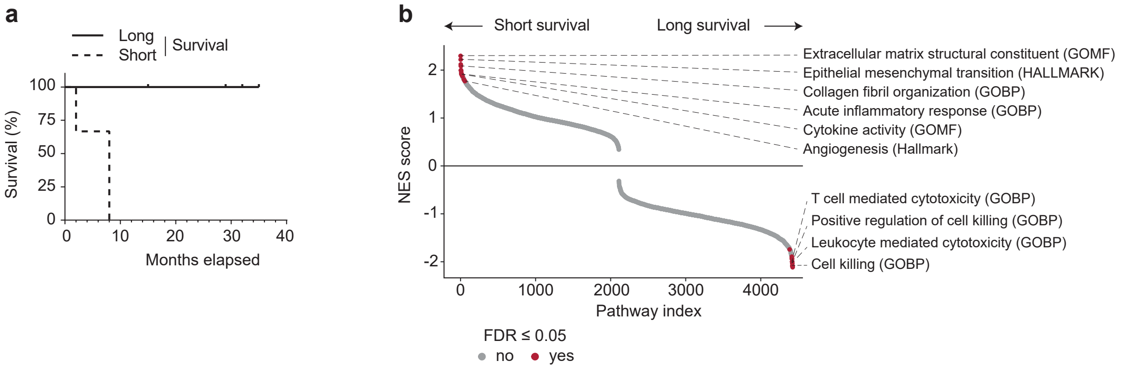

Figure 13: Expression patterns of genes in CD8$^+$ T cells associated with patient survival.
a, Kaplan-Meier survival curve comparing the survival estimation of patients since tumor resection grouped into short survival (below 1 year, n = 3) and long survival (above 1 year, n = 5). **b**, Gene set enrichment analysis of differentially expressed gene signatures in sorted tumor-infiltrating CD8$^+$ T cells identified by bulk RNA sequencing, divided into patients with short (below 1 year, n = 3) and long survival (above 1 year, n = 5) since tumor resection. NES scores of GSEA for each pathway are plotted, significantly enriched (p ≤ 0.05) pathways are highlighted in red. FDR, false discovery rate. (Tretter et al., 2023)

3.3 Genomic and transcriptomic data analysis

3.3.1 General characterization of the genomic and transcriptomic variant data

As one of the big "omics" data sets used in this study, the total genomic and transcriptomic variant data was analysed using self-developed R analysis scripts (Appendix 6.9.1). Here, the general characteristics of this data set and potential shared genomic features were assessed while the variant information was also used for the generation of a patient-specific data base enabling neoantigen identification later in this study (see 3.5), as already described in Tretter *et al.*.

First, the total number of genetic variants identified at the DNA and RNA level for each tumors sample was assessed. For a typical genomic data analysis, variants are filtered for several criteria such as coverage, variant frequency and number of mutated and wild type (wt) reads to reduce the identification of false positive variants. Applying standard filters for these criteria (coverage ≥ 5 reads, variant frequency ≥ 5%, mutated reads ≥ 2 in the tumor, mutated read ≤ 1 in normal control tissue), most genetic variants passed the filtering at RNA level for all tumor samples. However, there were several exceptions observed for variants found at the DNA level (Figure 14). In this thesis, all variants will serve as the foundation for neoantigen candidates identification and thus variants will subsequently be cross-validated with the MS-based tumor immunopeptidome data, the immunogenicity assessment and the post-validation steps (Figure 8). Hence, the decision was made to utilize the variant data sets containing unfiltered genetic variants, avoiding the loss of potential neoantigen candidates while acknowledging the associated increased risk of false positives.

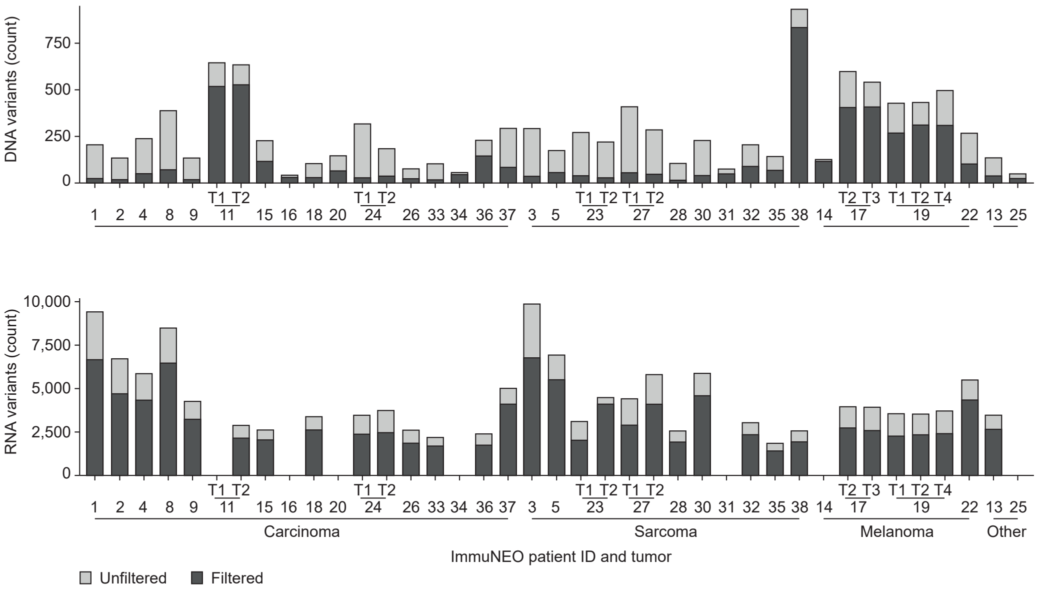

Figure 14: Quality assessment of variants identified at the DNA and RNA level.
The upper panel displays total unfiltered genetic variants (grey and black bars) and filtered genetic variants (black bars) identified by MuTect2 (v4.1.0.0) from whole exome (WES)/whole genome sequencing (WGS) data (DNA variants) per tumor sample, grouped by tumor entity. The lower panel shows unfiltered variants (grey and black bars) and filtered variants (black bars) identified by Strelka2 (v2.9.10) from RNA sequencing (RNA-seq) data (RNA variants) per tumor sample, also grouped by tumor entity. Filtering criteria included at a coverage of at least 5 reads, a variant frequency of 5%, and a minimum of 2 mutated reads within the tumor, with no more than 1 mutated read within normal control tissue. n = 39 tumor samples from n = 32 patients for WES/WGS data; n = 32 tumor samples from n = 26 patients for RNA-seq data (see Table 24). T, tumor. (Tretter et al., 2023)

The number of DNA and RNA variants exhibited considerable variability among patients, with no distinct differences observed across variable tumor entities in our pan-cancer cohort (Figure 15 a). On average, 302 DNA mutations per tumor were identified, whereas a substantially higher number of genetic variants was detected at the RNA level, averaging around 4024 variants per tumor (Figure 15 a). Notably, no discernible correlation between the quantities of DNA and RNA variants within each tumor was observed (Figure 15 b), suggesting that tumors with low somatic mutation levels, often classified as having low TMB (calculated based on the number of somatic mutations per megabase (Mb)), can still harbor a considerable amount of RNA variants. Also, more than half of the DNA variants were found at the RNA level (Figure 15 c), by that validating the identification of somatic mutations on a second level and also highlighting the power of RNA as a source for the general discovery of genetic variants. However, as some tumors lack RNA-seq data and gene expression into mRNA varies temporally, confirmation of some DNA variants might have been missed on RNA level.

In general, across both DNA and RNA levels, single-nucleotide substitutions predominantly constituted the observed variants, with occasional instances of deletions, insertions, and multi-nucleotide substitutions (Figure 15 d). Additionally, missense variants were the predominant variant type for both DNA and RNA variants, although RNA variants exhibited a higher proportion of splice-site and intron variants compared to DNA variants (Figure 15 e).

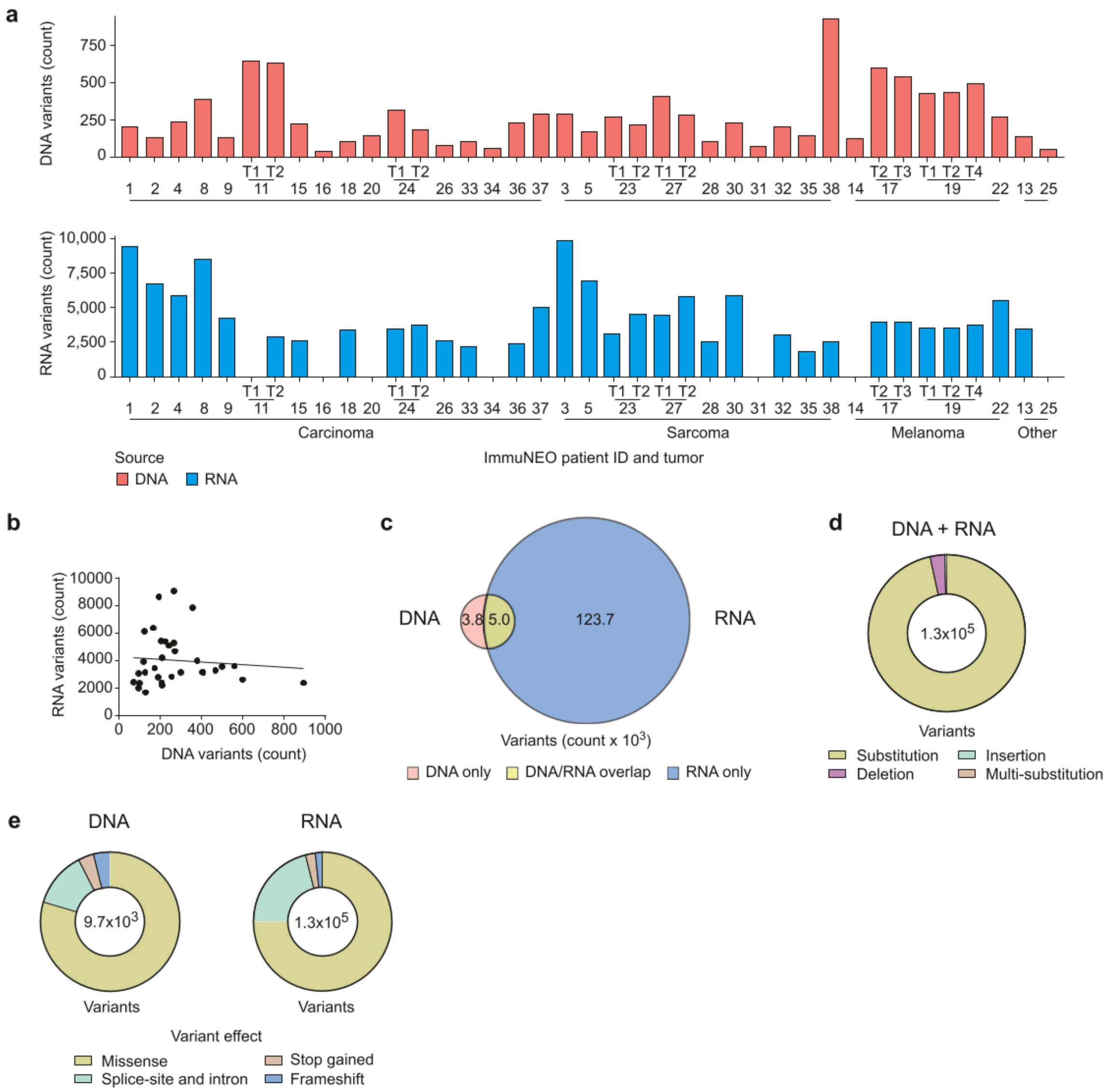

Figure 15: Identification and characterization of genetic variants identified in tumor tissue from different cancer entities at the DNA and RNA level.

a, Numbers of the total variants identified from DNA (upper panel) and RNA sequencing data (lower panel) per tumor sample, organized by tumor entity. Variants were called from whole exome (WES)/whole genome sequencing (WGS) data by MuTect2 (v4.1.0.0) and from RNA sequencing (RNA-seq) data by Strelka2 (v2.9.10), with SNP-filtering using the dbSNP-all data base. Note: RNA-seq data was unavailable for patients IN-11-T1, IN-14, IN-16, IN-20, IN-25, IN-31, IN-34. **b,** Correlation of the number of DNA and RNA variants in the same tumor sample, for patients where matching WES/WGS and RNA-seq data was available (n = 32 tumor samples). Symbols represent individual tumor samples. Spearman's rank correlation analysis resulted in ρ = 0.1578, with the linear regression shown in the graph (R^2=0.008). **c,** Venn diagram illustrating the overlap of unique variants identified from WES/ WGS data (DNA variants) and RNA-seq data (RNA variants). **d,** Proportions of each variant type for all identified genetic variants (total 1.325x10^5 variants), regardless of origin (WES/WGS and RNA-seq combined). **e,** Pie charts displaying the proportion of each variant effect from all DNA (left, 9.7x10^3 variants) and RNA variants (right, 1.287x10^5 variants). **a-e,** n = 39 tumor samples from n = 32 patients for WES/WGS data; n = 32 tumor samples from n = 26 patients for RNA-seq data (see Table 24). T, tumor. (Tretter et al., 2023)

The incorporation of RNA-seq data into the variant calling process expanded the data set to non-coding sources of variants such as regulatory RNAs and pseudogenes (Figure 16 a). Despite these additions, the higher count of variants detected at the RNA level compared to the DNA level was not entirely explained by these non-coding sources, as the majority of RNA variants originated from protein-coding regions (Figure 16 a). As already lined out in 1.2.2, the occurrence of RNA editing events might contribute to this elevated number of RNA variants (Bazak et al., 2014; Han et al., 2015). This possibility was further assessed by examining the variants exclusively identified at the RNA level. When checking the coverage at the corresponding wt locus at DNA level, a corresponding wt sequence was identified with min. 3 reads on DNA for most of the RNA variants (Figure 16 b). This suggest that these variants are not somatic and were not missed on DNA level. Instead, part of these variants may be attributed to RNA editing events. Moreover, a substantial portion of the RNA-only variants exhibited a characteristic A-to-G nucleotide exchange pattern, as associated with RNA editing events (defined as A-to-I editing, where I appears as G in RNA-seq data) (Bazak et al., 2014; Peng et al., 2018; Roth et al., 2019) (Figure 16 c).

Finally, the total genomic and transcriptomic data set was correlated to the patient's survival since tumor resection to identify potential common patterns. There was a positive trend observed in the correlation between TMB, measured by DNA variants per Megabase (Mb), and superior survival, although it did not reach statistical significance (Figure 16 d). Nevertheless, the overall number of DNA variants showed a positive correlation with prolonged survival in this heterogenous cohort (Figure 16 d), while there was no correlation identified between the number of genetic variants exclusively found at the RNA level and overall survival (Figure 16 d).

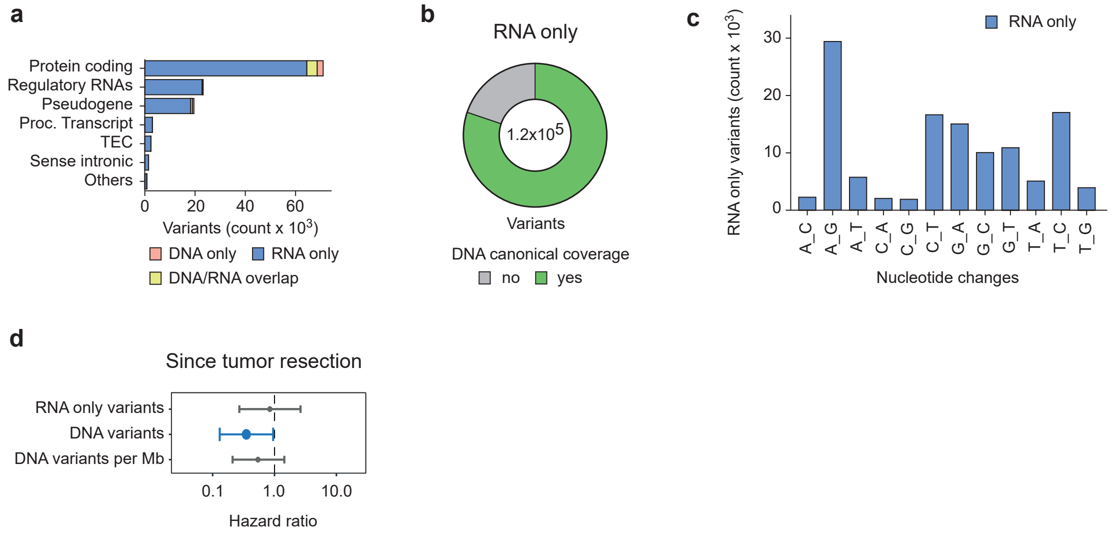

Figure 16: Origin and attributes of DNA and RNA variants and their association with patient survival.
a, Bar graph illustrating the number of variants in each genetic biotype and the originating dataset. **b,** Pie chart displaying the proportion of variants across all tumor samples exclusively identified from RNA sequencing (RNA-seq) data, where the respective canonical sequence was identified at the DNA level with a coverage of ≥ 3 reads (green) or the region was not covered (grey, < 3 reads) **c,** Distribution of the nucleotide exchange pattern across all single nucleotide variants exclusively identified from RNA-seq data. **d,** Forest plot depicting the hazard ratio (dot) and 95% confidence intervals (lines) calculated by log-rank test and Cox's proportional hazards model for several genetic parameters affecting the patient's survival since tumor resection (DNA variants n = 32 patients, RNA variants n = 26 patients). Significant results ($p \leq 0.05$) are highlighted in blue. Only one representative tumor sample per patient was used for statistical testing (see core cohort Table 24). **a,d,** n = 39 tumor samples from n = 32 patients for WES/WGS data; n = 32 tumor samples from n = 26 patients for RNA-seq data. **b,c,** n = 32 tumor samples from n = 26 patients. Mb, megabase; Proc., processed; T, tumor; TEC, to be experimentally confirmed; wt, wild type. (Tretter et al., 2023)

3.3.2 Identification of shared pan-cancer variants and mutational patterns

In addition to characterizing the genomic and transcriptomic variant data used for neoantigen identification, this study aimed to assess potential shared genetic variants within this pan-cancer cohort. Such shared variants could give rise to common TAAs or TSAs, making them attractive targets for immunotherapy. Therefore, it was investigated if there was an overlap of variants between all analyzed patients and in how many patients each genetic variant was detected. The resulting data was already published in Tretter *et al.*. The investigation revealed that the majority of genetic variants were unique in the cohort (Figure 17 a and b), with approximately 97% unique variants at the DNA level (Figure 17 a) but only 89% at the RNA level (Figure 17 b). Considering that overall roughly 10 times more RNA variants were detected than DNA variants, approximately 37 times more shared genetic variants (detected in at least 2 patients) were detected at the RNA level. Notably, a subset of RNA variants was found to be shared among all patients (n = 26), while DNA variants were shared in smaller patient groups and less frequently (Figure 17 a and b).

Therefore, the shared RNA variants were assessed in more detail to understand if these shared RNA variants were not only shared by the same patients but if groups of variants were found together on a sample level and in the same sets of patients. The Upset plot method was employed to investigate shared RNA variants found in at least ten tumor samples with a minimum of two shared RNA variants (Figure 17 c). While the majority of shared RNA variants in these sets were exclusive, the analysis revealed 59 shared variants with some degree of overlap (Figure 17 c). Among these, 11 RNA variants were consistently present in all patients and tumor metastases within the pan-cancer cohort (Figure 17 b and c). Notably, overlapping shared RNA variants were not confined to tumor metastases of the same patient but were also identified in different tumor entities across the cohort.

Taken together, remarkably more genetic variants in general and shared variants in particular were identified at the RNA level, and a substantial part of additional RNA variants was likely derived from events happening on RNA level only.

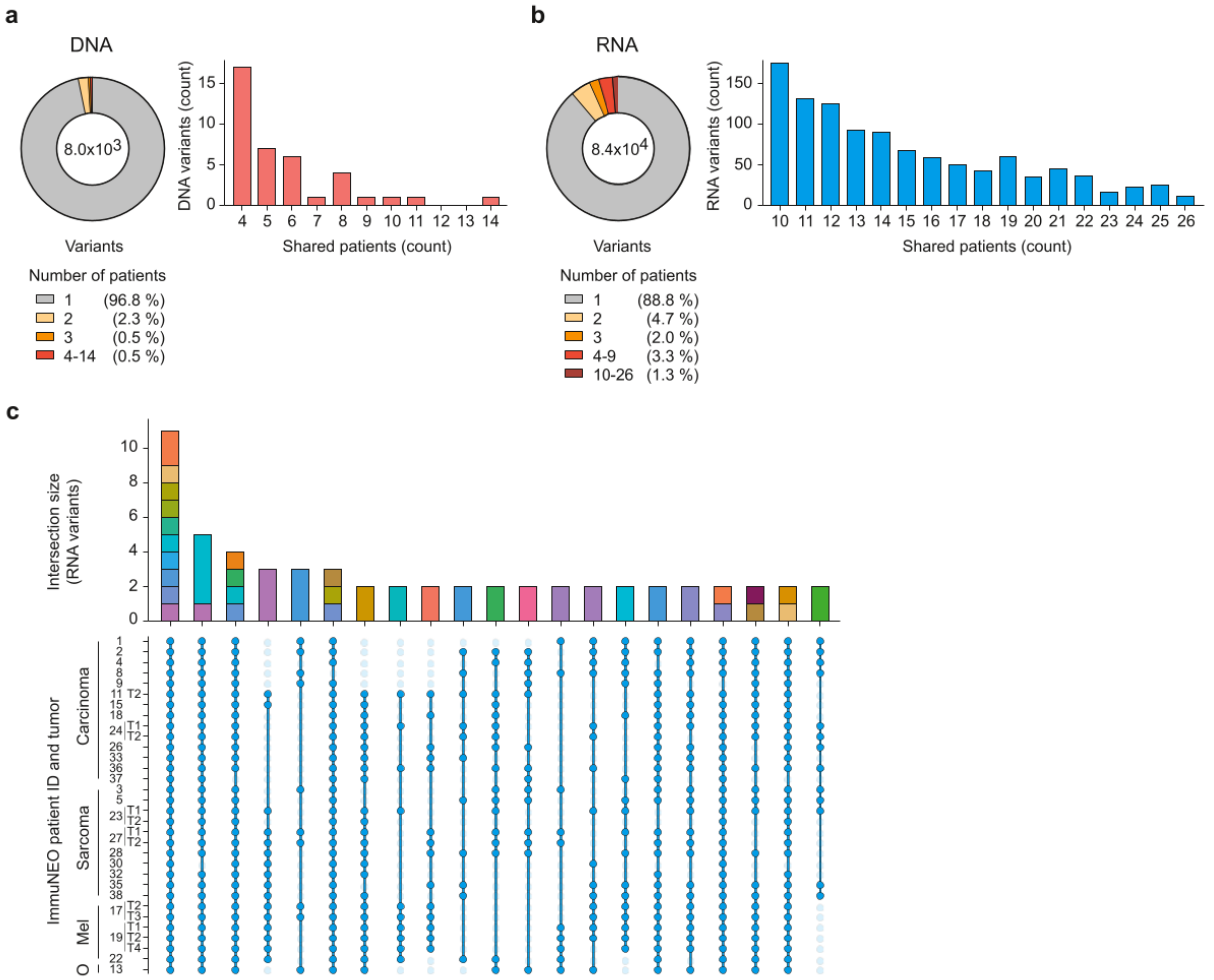

Figure 17: Shared genetic DNA and RNA variants.
a,b, Pie charts illustrating the ditribution of unique and shared DNA (**a**) and RNA variants (**b**) across different patients. The adjacent bar graphs detail the count of variants shared by 4 to 14 patients for DNA variants (**a**) and shared by 10 to 26 patients for RNA variants (**b**). **c,** Upset plot displaying the overlap of at least 2 RNA variants among at least 10 tumor samples. The bar graph indicates the number of unique variants found in the same subset of tumors (intersection size), with dots representing tumor samples within the subset and lines connecting samples within the same subset. Different genes containing the specific variant are colour-coded in the intersection bar graph. **a-c,** n = 39 tumor samples from n = 32 patients for WES/WGS data; n = 32 tumor samples from n = 26 patients for RNA-seq data (see Table 24). Mel, melanoma; O, other T, tumor. (Tretter et al., 2023)

3.4 Characterisation of tumor immunopeptidomes

3.4.1 General characterization of immunopeptidomc data

As a second "omics" data set and input data for the identification of neoantigens presented on tumor cell surface, the HLA-I immunopeptidome of each tumor sample was assessed in this pan-cancer cohort. Therefore, immunoprecipitation of pHLA-I followed by MS analysis was performed as previously described (Bassani-Sternberg et al., 2016). Subsequently all measured peptide sequences (immunopeptidome) were identified by matching the MS spectra to the total wild type proteome using pFIND (Chi et al., 2018) and an analysis script for the analysis of this big data set using R was developed (see Appendix 6.9.2). The resulting data was already published in Tretter *et al.*.

Similar to the distribution of genetic variants, there was considerable variability in the overall numbers of peptides among patients, with no apparent difference observed between distinct tumor entities (Figure 18 a and b). On average, approximately 5075 peptides were identified per tumor (Figure 18 a), ranging from 346 to 15,983. However, the distribution pattern changes when quantifying the numbers of identified peptides per gram of tumor tissue.

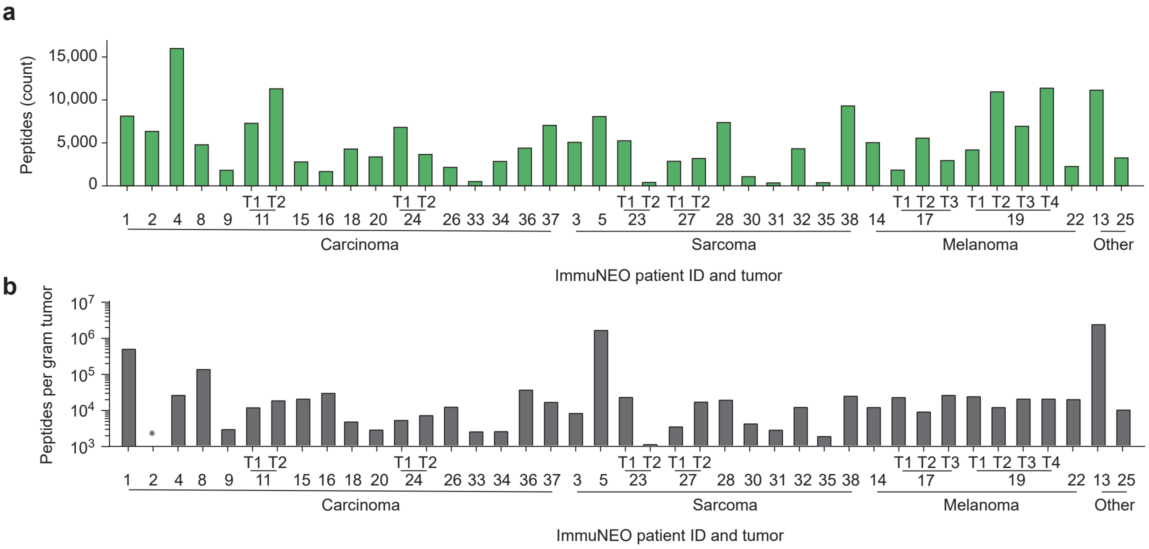

Figure 18: Identification and quantification of the HLA class I tumor immunopeptidome per patient.
a, Presentation of the overall count and **b,** the quantified count per gram tumor of unique HLA class I peptides identified per tumor sample, sorted by tumor entity. The isolation of peptides bound to HLA class I molecules on tumor cell surfaces was performed by immunoprecipitation, followed by sequencing through liquid chromatography with tandem mass spectrometry (LC-MS/MS). The identified peptide sequences were subsequently aligned to the Ensemble92 protein database using pFIND (v3.1.5) with 1% FDR. Unique sequences were filtered and counted per tumor sample. n = 41 tumor samples from n = 32 patients (see Table 24). T, tumor. (Tretter et al., 2023)

For quality assessment, the length distribution of the identified peptides per patient was assessed resulting in a range of 8 to 15 amino acids in length, while predominated by nonamers (Figure 19). Furthermore, the HLA anchor residues of the immunopeptides were analyzed in all patients using MHCMotifDecon (v1.0) (Kaabinejadian et al., 2022) and it thereby could be shown that in mean 95% of all peptide sequences were characteristic for the respective patients' HLA composition and only 5% of identified peptides could not be matched (exemplified in four patients ImmuNEO-4, -11, -14, -38, Figure 20, mean percentage of HLA-assigned peptides over all patients 96.4%). This highlights the purity of the data set and gives a good understanding of the quality.

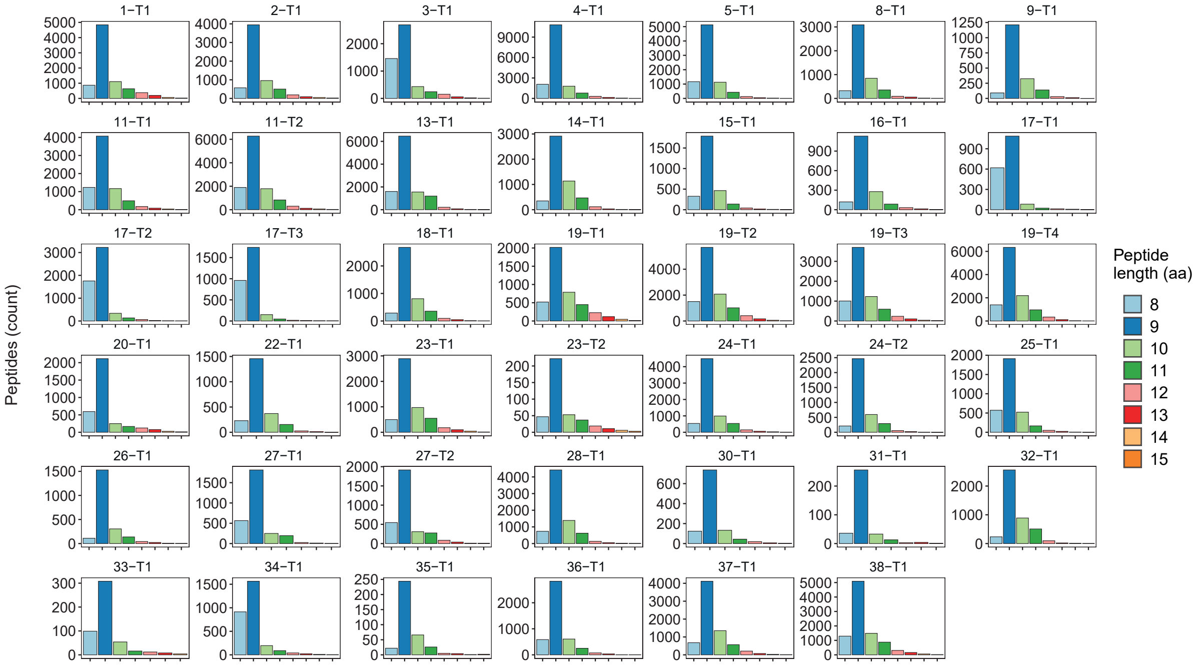

Figure 19: Length distribution of eluted HLA class I peptides identified through mass spectrometry.
Bar graph illustrating the distribution of unique peptides across various peptide lengths measured in amino acids for each tumor sample. The isolation of peptides bound to HLA class I molecules on tumor cell surfaces was performed by immunoprecipitation, followed by sequencing through liquid chromatography with tandem mass spectrometry (LC-MS/MS). The identified peptide sequences were subsequently aligned to the Ensemble92 protein database using pFIND (v3.1.5) with 1% FDR. Unique sequences were filtered and counted per tumor sample. n = 41 tumor samples from n = 32 patients (see Table 24). aa, amino acids; T, tumor. (Tretter et al., 2023)

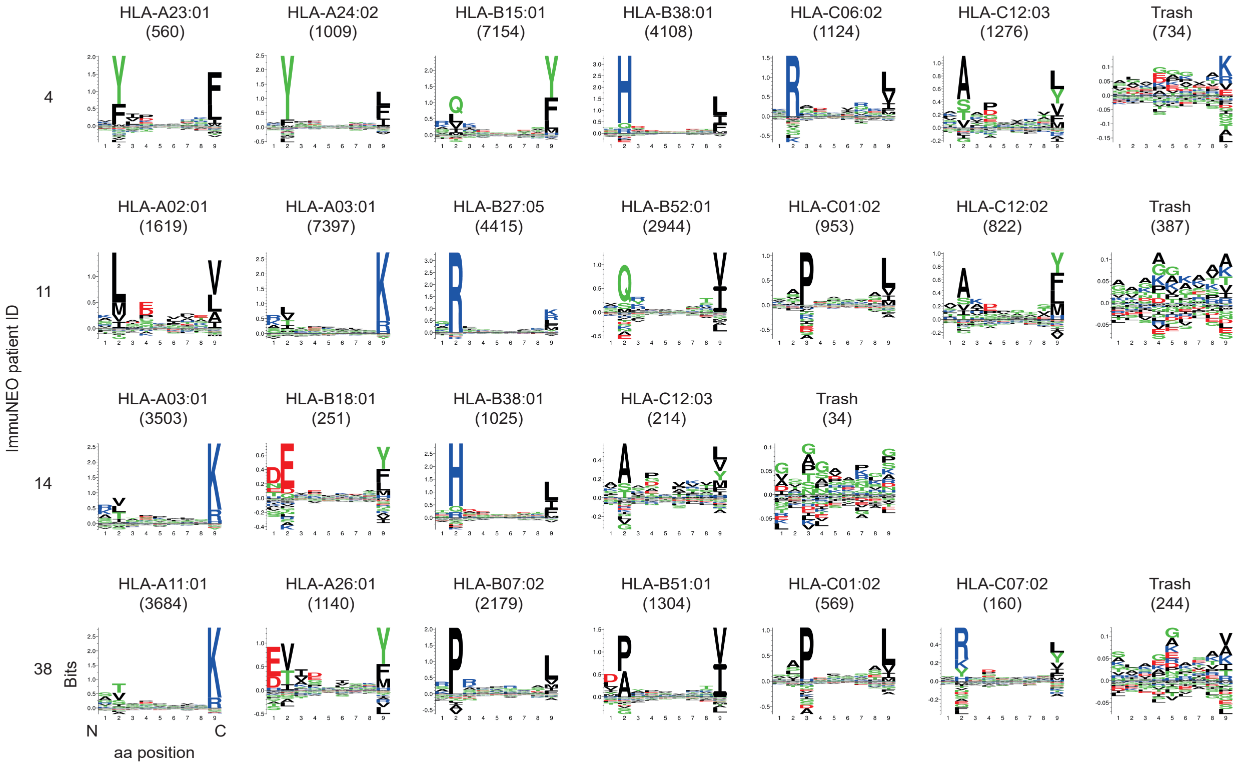

Figure 20: HLA class I binding motifs of peptides within the immunopeptidome of selected ImmuNEO patients. MHCMotifDecon (v1.0) was employed to align HLA class I peptides ranging from 8-15 amino acids in length to the patients' specific HLA class I alleles based on their binding motifs and anchor residues. The binding motifs for four representative tumor samples per HLA class I allele are presented, including the total count of matched peptide sequences in parentheses. Peptides that did not match any HLA class I allele of the respective patient are indicated in the trash subgraph. aa, amino acid; HLA, human leukocyte antigen. (Tretter et al., 2023)

3.4.2 Identification of shared pan-cancer peptides with potential immunogenic function

For the detailed analysis of the immunopeptidomic data set we focused on the identification of tissue-agnostic shared peptides and patterns that might be relevant for immunotherapy approaches. Therefore, we looked at peptides origination from cancer-associated genes that have been described in the Human Protein Atlas (Uhlen et al., 2017) and at peptides arising from known CTAs published in the CTpedia database (CTpedia, 2021). The resulting data was already published in Tretter *et al.*.

When comparing peptides derived from cancer-associated genes, it was noticed that 36% of these peptides were shared among at least two patients (Figure 21 a), with a notable number present in up to 18 patients (Figure 21 b). Among these shared peptides, 79 exhibited some degree of overlap not only between patients but also across at least eight distinct tumor samples in groups ranging from 2 to 15 peptides (Figure 21 c). Additionally, 18 shared peptides were identified in at least 11 patients (Figure 21 c marked by arrows). Intriguingly, all patients harboring these 18 shared peptides shared common HLA-A molecules (HLA-A03:01 or HLA-A11:01; see Table 6) and according to NetMHC4.0, all shared peptides were predicted to bind with good affinities (< 200 nM) to these two HLA molecules. Furthermore, these peptide ligands have been previously described in the context of cancer by multiple studies (IEDB.Org: Free Epitope Database and Prediction Resource, 2022; PeptideAtlas, 2022).

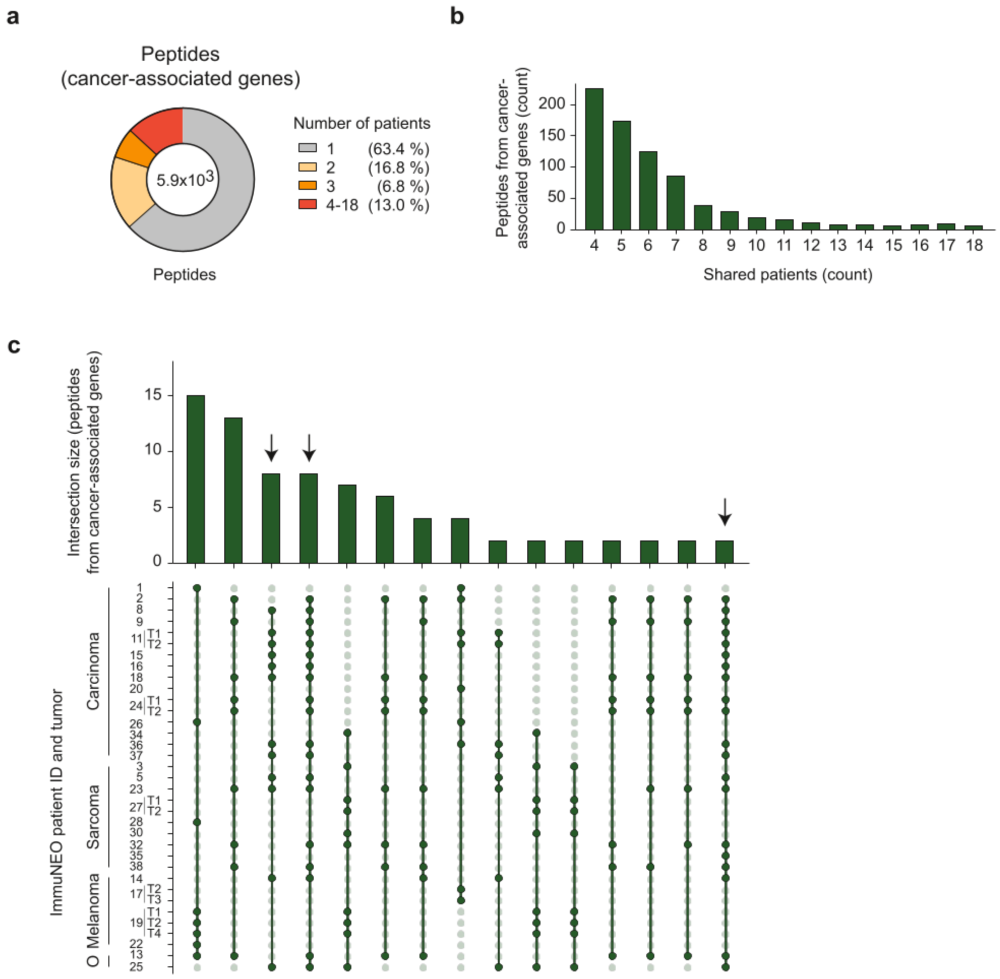

Figure 21: HLA class I peptides that are common or shared across the pan-cancer cohort.
a, Pie chart illustrating the distribution of peptides originating from cancer-associated genes (ProteinAtlas), highlighting the proportion that is unique to individual patients and those that are shared. **b,** Bar graph providing a detailed breakdown of the number of peptides shared among 4 to 18 patients. **c,** Upset plot displaying the overlap of unique peptides from tumor-associated genes, as annotated by the Protein Atlas, between all tumor samples. The bar graph indicates the count of unique peptides found in the same subset of tumors (intersection size), with dots representing tumor samples within the subset and lines connecting samples within the same subset. Subsets of peptides present in at least 11 patients are emphasized with arrows. **a-c,** n = 41 tumor samples from n = 32 patients (see Table 24). O, other; T, tumor. (Tretter et al., 2023)

In addition, when analyzing peptides derived from reported CTAs, numerous CTA peptides were discovered in our cohort as illustrated in the heatmap in Figure 22 a. While the majority of peptides originating from CTAs were unique to one patient (Figure 22 a lower part), multiple CTA genes were identified not only to generate several presented peptides but also to be present in a substantial portion of patients, irrespective of the tumor entity (e.g. ATAD2, SPAG9, ODF2, KIAA0100) (Figure 22 a upper part). Moreover, there was not only an overlap between the same CTA genes inducing peptides across different patients, but several CTA peptides were identified in multiple patients (Figure 22 b). For instance, a peptide originating from cTAGE5 was shared by 12 patients and several peptides from ATAD2 were found in 10 patients.

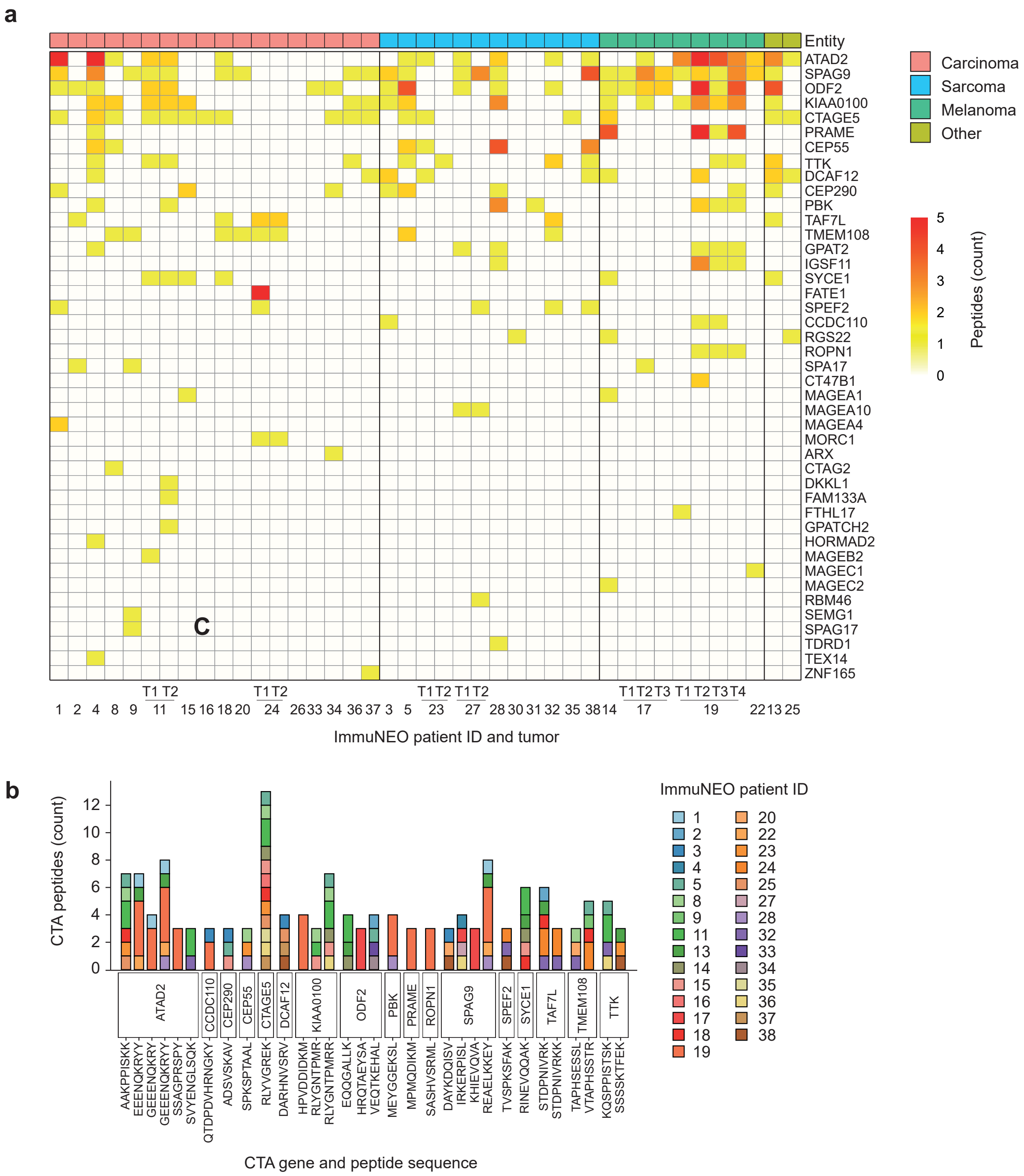

Figure 22: Shared pan-cancer peptides originating from cancer testis antigens.
a, Heatmap illustrating the count of distinct peptides identified from each cancer testis antigen (CTA) gene in individual tumor sample. Genes are arranged based on the cumulative peptide count across all patients (highest to lowest count), and samples are grouped by tumor entity. **b,** Bar graph providing a detailed breakdown of all shared peptides derived from CTA genes, including their respective sequence and the patients in which they were identified. n = 41 tumor samples from n = 32 patients (see Table 24). CTA, cancer testis antigen; T, tumor. (Tretter et al., 2023)

In conclusion, the comprehensive analysis of the immunopeptidome in this cross-entity cohort revealed high quality data and, more importantly, let to the identification of several potential tumor-associated antigen candidates for immunotherapy in a shared patient cohort.

3.5 Identification of patient-specific neoantigens

3.5.1 Neoantigen identification by proteogenomics

The center piece of this study was the identification of neoantigen candidates using proteogenomics, which combines the previously characterized genomics, transcriptomics and immunopeptidomics data sets. We have further optimized the previously published bioinformatics pipeline (Bassani-Sternberg et al., 2016) within this study. Several novel tools were implemented into the pipeline such as the use of transcriptomic data for mutation calling, which increased the search space for potential neoantigen candidates tremendously (see 3.3.1). Also, the mutation calling algorithm (Lange et al., 2020) was expanded and the mutation to peptide converter VCF-translate (Tretter et al., 2023) was improved (see 2.2.5.1). Furthermore, the peptide identification algorithm pFind (Chi et al., 2018) was used in contrast to MaxQuant enabling faster sequence-spectra matching within this large search space (see 2.2.5.2). To increase the potential for neoantigen candidate detection even further, the machine learning algorithm Prosit (Gessulat et al., 2019) was integrated as a second neoantigen calling algorithm into the pipeline (see 2.2.5.2). Subsequently, identified neoantigen candidates had to pass our comprehensive and extensive post-processing pipeline to minimize identification of false positives, which was developed by myself together with Philipp Seifert and which is described in detail in the method section (see 2.2.5.3). The resulting data was already published in Tretter *et al.*.

Using the improved proteogenomic pipeline, a total of 90 neoantigen candidates were successfully identified. Of these, 77 neoantigens were discovered using pFIND, and an additional 14 neoantigens were identified through the Prosit-based rescoring approach (Figure 23 a). These 90 neoantigen candidates were found in 24 patients across different tumor entities (75% of all patients and 88% of patients with available RNA-seq data), emphasizing the presence of potential targets for personalized immunotherapy irrespective of their cancer type. Per patient, the number of neoantigen candidates varied from 1 to 13 (Figure 23 b, Appendix 6.3). The peptide lengths in amino acids of all identified neoantigen candidates varied from 8 to 14-mers, with nonamers being the most prevalent (Figure 23 c).

Most strikingly, 79 out of the 90 identified neoantigen candidates were exclusively derived from RNA variants. In contrast, 3 neoantigen candidates originated solely from DNA variants, and eight were shared between both sources (Figure 23 d). In most cases of neoantigen candidates exclusively derived from RNA variants, a corresponding canonical sequence was detected at the DNA level (Figure 23 e) aligning with observations made for the overall numbers of RNA-only variants (see 3.3.1). Moreover, many of these variants also harbored the RNA editing associated nucleotide exchange pattern (A-to-G) (Figure 23 f). Furthermore, a substantial amount of neoantigen candidates was derived from non-coding regions such as pseudogenes and lncRNAs (Figure 23 g, right), highlighting the importance of other genomic regions as sources for neoantigens. Concerning the variant effect of the variants associated with the identified neoantigen candidates, missense variants remained the most abundant. However, splice-site and intron variants were enriched (Figure 23 g, left) compared to their proportion in the overall variant distribution (compare to Figure 15 e).

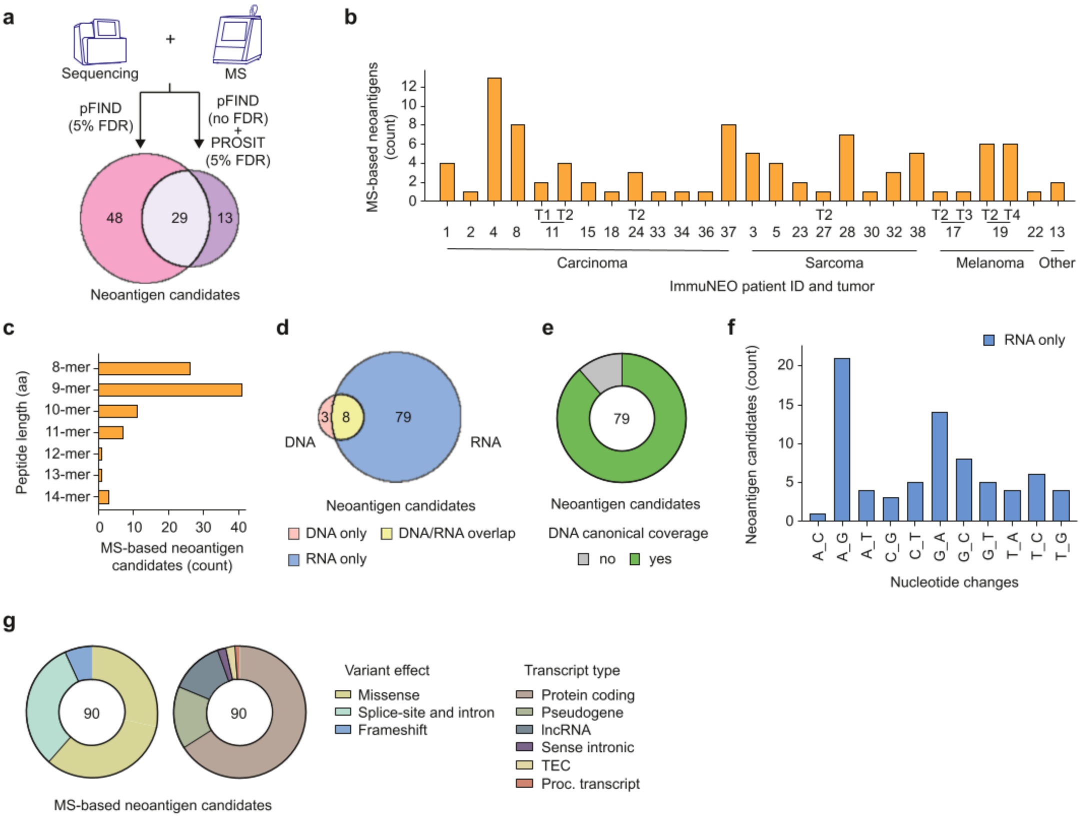

Figure 23: Identification and characterization of neoantigen candidates through proteogenomic analysis.
a, Number of neoantigen candidates based on the bioinformatics tool used for their identification and the overlap. pFIND (v3.1.5) (Chi et al., 2018) was utilized at a false discovery rate (FDR) of 5% on the spectral level for identifying non-canonical 8-15mer neoantigen candidates. The machine learning tool Prosit (Gessulat et al., 2019) was additionally integrated for neoantigen identification, using unfiltered pFIND data for rescoring of peptide spectra matches to the patient-specific ORF database. A total of n = 39 tumor samples from n = 32 patients were analysed and n = 27 tumor samples from n = 24 patients harboured n = 90 neoantigen candidates. **b,** Number of identified neoantigen candidates per tumor sample, sorted by tumor entity. **c,** Length distribution in amino acids (aa) of all identified neoantigen candidates. **d,** Source (DNA or RNA data) of the variants from which the neoantigen candidates originated. **e,** Proportions of neoantigen candidates identified exclusively from RNA sequencing (RNA-seq) data, distinguishing cases where the respective canonical sequence was identified at the DNA level with a coverage of ≥ 3 reads (green) and where the respective region was not covered (grey, < 3 reads). **f,** Representation of the nucleotide exchange pattern of all single nucleotide variants generating neoantigen candidates identified exclusively from RNA-seq data. **g,** Proportions of each variant effect type (left) and transcript biotype (right) of all variants inducing neoantigen candidates. **a-g,** n = 39 tumor samples from n = 32 patients were analysed in total; n = 27 tumor samples from n = 24 patients harboured n = 90 neoantigen candidates; n = 3 neoantigen candidates from DNA variants; n = 8 neoantigen candidates from DNA and RNA variants; n = 80 neoantigen candidates from RNA variants. aa, amino acids; MS, mass spectrometry; Proc., processed; T, tumor; TEC, to be experimentally confirmed. (Tretter et al., 2023)

Of note, we again looked for common targets and shared neoantigen. No shared neoantigen candidates were identified across different patients. Nevertheless, three peptides were found to be shared between two metastases in a melanoma patient (ImmuNEO-19) and one peptide was shared between two separate tumor samples of different entity in a patient with dMMR (ImmuNEO-11) (Appendix 6.3).

When looking for neoantigen candidates arising from previously identified shared genetic variants (see 3.3.2), two neoantigen candidates in two distinct patients (ImmuNEO-4 and -23) were identified that were derived from shared variants in MAP4K5 (IN_04_F, identified with 1.5% FDR; shared between 32 tumor samples; Appendix 6.3) and in AC024075.2 (IN_23_A, identified with 4.3% FDR, shared between 24 tumor samples; Appendix 6.3), respectively. When specifically investigating the unfiltered pFind files for mutated peptides from the shared alterations irrespective of their FDR, one additional mutated peptide ligand was identified with FDRs of 5.6% and 8.1% in ImmuNEO-03 as a result of a shared RNA alteration in WASHC2A. Several additional peptides were identified by MS harboring shared variants (somatic and RNAonly) although the FDR was comparably high in most of them (75% of mutated peptides had FDRs from 22-63%).

In conclusion, this data indicates that the MS-based identification of potential neoantigens is feasible in most cancer patients, regardless of tumor entity. Additionally, tumor transcriptomic data serves as a crucial source for detecting peptide ligands derived from genetic variants.

3.5.2 Neoantigen identification by in silico prediction

In addition to the proteogenomic-based neoantigen identification, a de-novo *in silico* prediction approach was followed using the mutation calling data from section 3.3 in combination with NetMHC4.0, a method often used as a gold standard for neoantigen identification. As this data analysis would take all detected genetic variants into account, which would exceed the computational and analytical capacity of this project, the prediction was focused on only SNVs and nonameric mutated peptides as described in 2.2.6.

This approach resulted in the prediction of a large amount of potential neoantigen candidates in all 32 analysed patients (in total 28154 peptides). The numbers of predicted neoantigens varied per patient and tumor (Figure 24 a), but exceeded by far the amount of neoantigen candidates identified via MS (91 in total). Also, only seven MS nonameric peptides could be confirmed by the prediction approach (Figure 24 b). As a side note, some samples have very low numbers of predicted neoantigen candidates, as here no RNA-seq data and thus only few variants were available for this analysis.

Furthermore, several potential 9mer neoantigen candidates were identified from the previously described shared DNA (Figure 24 c) and RNA variants (Figure 24 d) in several patients, that have not been found by MS. Two genes, POLRMTP1 and MAP4K5, seem to not only be mutated in 10 samples (DNA data) and 32 samples (RNA data) respectively, but also lead to several predicted high-binding neoantigen candidates in multiple patients (11 patients and 16 patients, respectively).

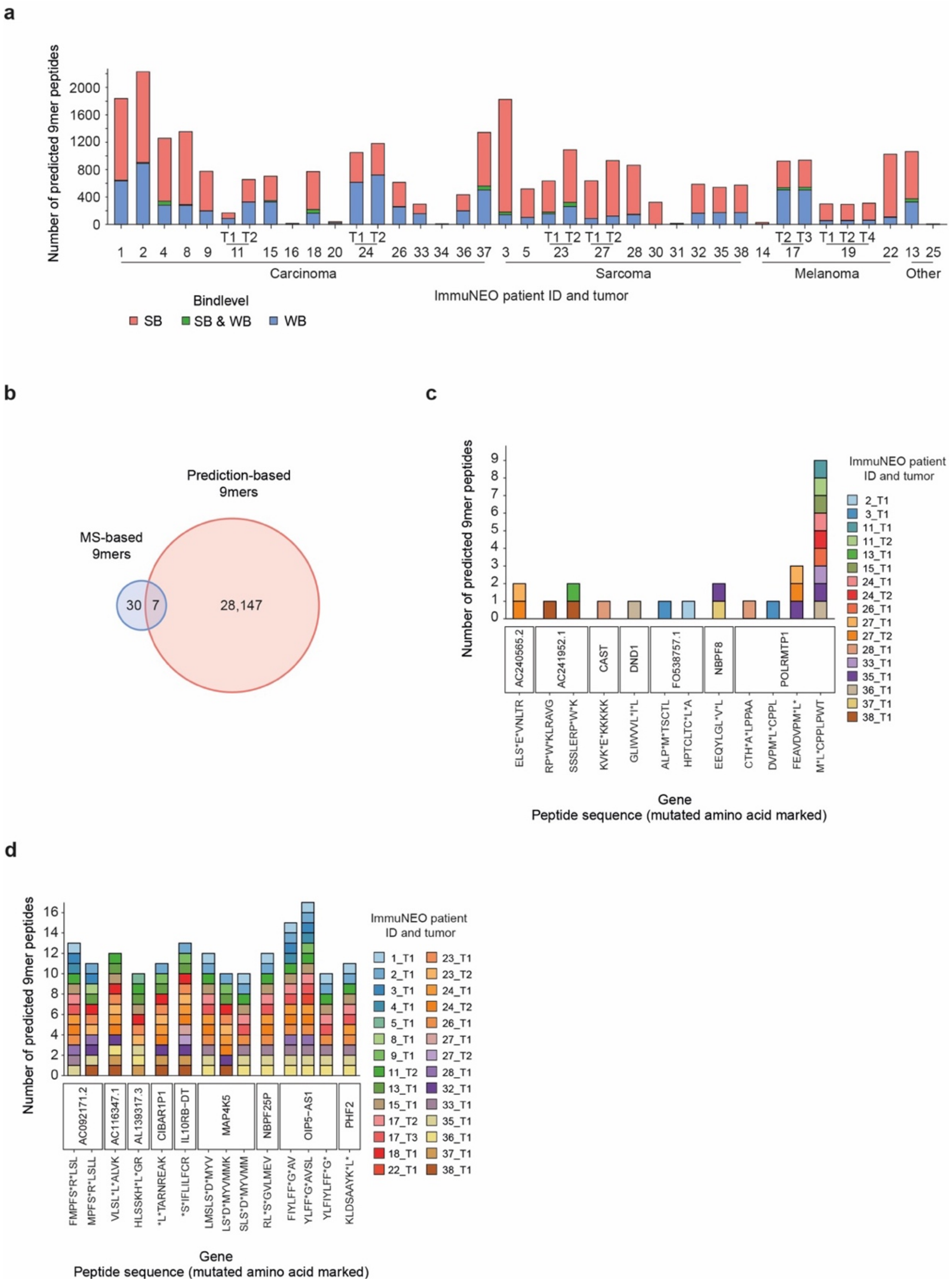

Figure 24: *In silico* prediction of nonameric neoantigen candidates.

a, Predicted neoantigen candidates per tumor sample. From the genomic mutation data (DNA and RNA, SNVs only) nonameric mutated peptide ligands were predicted using NetMHC4.0. Peptides classified by rank as strong (SB) or weak binders (WB) and predicted binding affinities <200nM were filtered as neoantigen candidates and unique peptides per tumor sample are shown. **b,** Venn diagram showing the number of nonameric neoantigen candidates identified via the MS pipeline and the prediction pipeline as well as their overlap. **c,** Precited neoantigen candidates from shared somatic mutations (shared by min. 4 patients, Figure 17 a) are shown, the corresponding gene and the originating tumor samples are annotated. **d,** Precited neoantigen candidates from shared RNA alterations (min. 10 unique samples, Figure 17 c) were extracted and are shown for peptides found in at least 10 samples. The corresponding gene and the originating tumor samples are annotated. **a-d,** n = 39 tumor samples from n = 32 patients. MS, mass spectrometry; SB, strong binder; SNV, single nucleotide variation; T, tumor; WB, weak binder.

3.5.3 Comparison to Mel15

To evaluate the performance of the improvement analysis pipeline, the previously published patient Mel15 (Bassani-Sternberg et al., 2016) was re-analyzed with the improved mutation calling and neoantigen identification pipeline including also RNA-seq data and the extensive post-processing procedure. This resulted in the detection of substantially more total variants (56,808 DNA and RNA variants) within this patient than published before (3965 DNA variants). Also, the total number of identified neoantigen candidates for Mel15 was increased by 38-fold to 307 mutated peptide ligands by applying our new pipeline (8 neoantigen candidates in the previous publication). In comparison to the ImmuNEO cohort, this melanoma patient seems to be an exception in regard to mutations, number of total peptides and specifically neoantigen candidates (see Table 25), also when only looking at melanoma patients within this cohort.

Table 25: Comparison of the ImmuNEO data sets to Mel15.
The raw data sets of Mel15 tumor 1 were re-analysed using the improved analysis pipeline described in this study also including RNA sequencing data for mutation calling and neoantigen identification with post-processing. Results were compared to the ImmuNEO data.

Dataset		ImmuNEO mean	Mel15	Ratio
Genomic data	DNA variants	302 (n=39)	3,161 (n=1)	10.5x
	RNA variants	4,024 (n=32)	53,647 (n=1)	13.3x
Immunopeptidome	-	5,075 (n=41)	34,236 (n=1)	6.7x
Neoantigen candidates	-	2.4 (n=39)	304 (n=1)	126.6x

3.6 Immunogenicity assessment of neoantigen candidates

3.6.1 Identification and description of immunogenic neoantigens

As a validation of the identified neoantigen candidates and to potentially isolate neoantigen-specific T cells for future T-cell based therapies, the immunogenicity of the mutated peptide ligands was assessed using *in vitro* stimulation assays. Therefore, the methodology of the previously described acDC assay (Bassani-Sternberg et al., 2016; Martinuzzi et al., 2011b) was used and slightly modified within this work. The resulting data was already published in Tretter *et al.*.

The modified acDC assay was then used, either with or without CD137^{+}-enrichment, for the immunogenicity assessment of the neoantigen candidates. T cell responses against 78 neoantigen candidates from 21 patients were evaluated in *in vitro* assay using autologous PBMCs from different blood drawl time points or *in vitro* expanded TILs by ELIspot analysis (see Figure 6). Furthermore, acDC assays using allogenic HLA-matched PBMCs was performed for ten of the identified neoantigen candidates.

Representative results for three IFN-γ -ELIspot assays are shown in Figure 25 a. The immunogenicity of a neoantigen was defined by the spot counts at day 13, comparing the mean spots from the mutated peptide condition against those of the control peptide condition. In this study, reactivity/positive response was considered when the ratio exceeded 2, signifying that the mutated peptides induced an IFN-γ response in at least twice as many T cells compared to the control. Additionally, a difference of spots above 50, defined as the background threshold for unspecific stimulation, was taken into account. Based on these criteria, the acDC data from 13 experiments were assessed and summarized in Figure 25 b and c (data provided in Appendix 6.4).

Out of 78 examined neoantigen candidates, 21 demonstrated the ability to induce T cell responses, accounting for 27% of all tested neoantigen candidates. These responses were observed in various experimental settings, including the use of autologous PBMCs (Figure 25 b, left), expanded TILs (Figure 25 b, right), or allogenic-matched PBMCs (Figure 25 c) (reactive peptides marked in Appendix 6.3). Most immunogenic neoantigen candidates were found when using autologous PBMCs, while only three immunogenic neoantigens were detected from expanded autologous TILs (Figure 25 b). Furthermore, a set of neoantigen candidates (n=10) was tested using healthy donor allogenic-matched PBMCs. These assays confirmed immunogenicity for four neoantigen candidates previously identified as immunogenic in the autologous setting and discovered one additional immunogenic neoantigen (IN_19_A) (Figure 25 c). Notably, immunogenic neoantigens were not preferentially identified by either of the two processing workflows pFind and Prosit nor by both of them (Figure 25 d, Appendix 6.3).

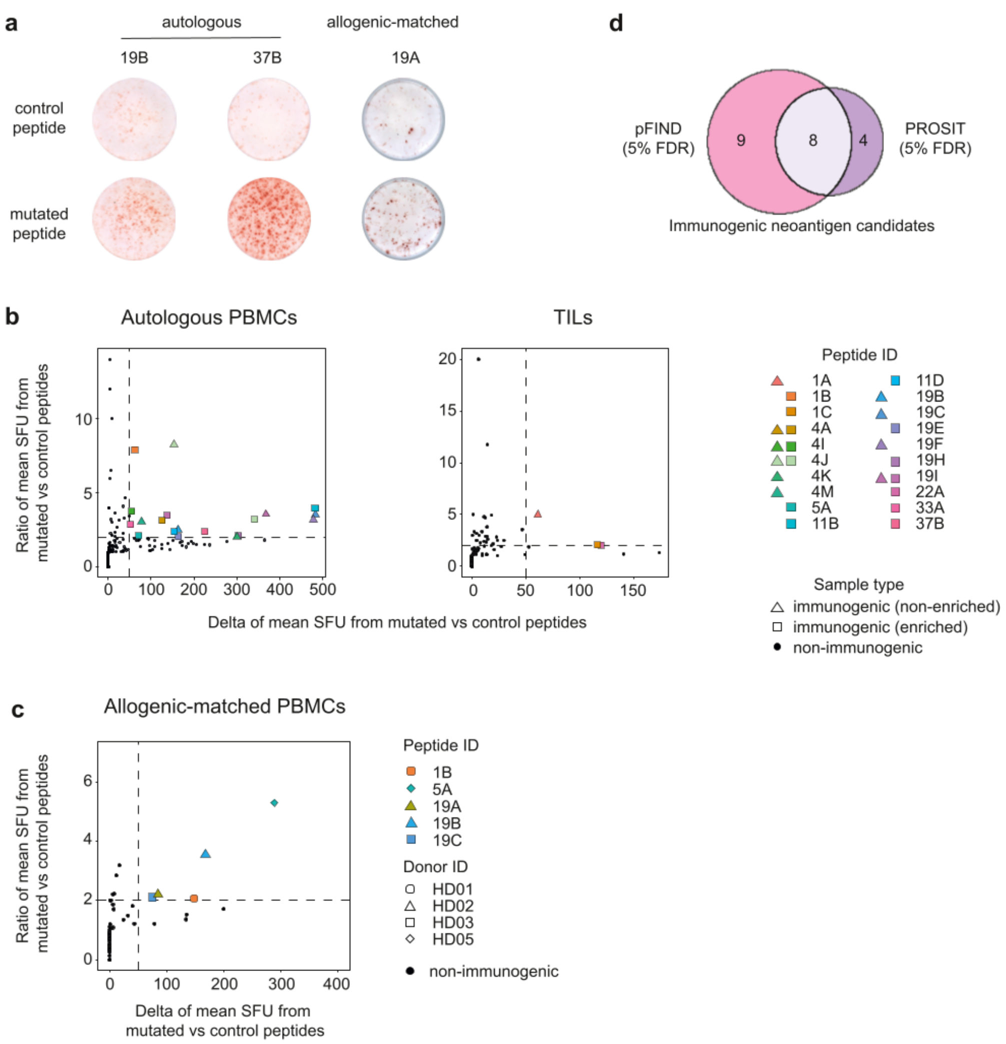

Figure 25: Immunogenicity assessment of neoantigen candidates.

a, Representative data of IFN-γ ELIspot assays displaying spot forming units (SFU) per well, comparing an irrelevant control peptide (top) with the indicated neoantigen candidate peptide (bottom). Assays using autologous as well as allogenic-matched PBMCs are shown. **b, c,** Summary of immunogenicity assessment data from all conducted modified accelerated co-cultured dendritic cell (acDC) assays for neoantigen candidates using ELIspot analysis. This included patient-derived PBMCs (non-enriched – left plot, CD137^{+} enriched – middle plot) or TILs (enriched and non-enriched combined – right plot) (**b**) as well as allogenic-matched healthy donor PBMCs (non-enriched) (**c**). Mean IFN-γ SFU for T cells tested against the neoantigen candidate peptide (test condition) and an irrelevant control peptide (control condition) were determined, and the ratio as well as the difference of the mean SFU of both conditions was calculated. Respective values are plotted for each peptide and PBMC or TIL aliquot tested. Peptides inducing an immune response, defined as a SFU ratio of > 2 and a difference of > 50, are highlighted. Autologous lymphoblastoid cell lines (LCLs) or allogenic HLA-matched cell lines (LCLs or HLA-transduced cell lines) were utilized as target cells. For better readability, negative values (when the control condition displays more spots than the test condition) were set to 0. **d,** Number of immunogenic neoantigen candidates identified by each algorithm and their overlap. **b, d** n = 78 neoantigen candidates from n = 24 patients were analysed in total; n = 8 patients harboured n = 20 immunogenic neoantigens; n = 17 immunogenic neoantigen candidates from autologous PBMC cultures; n = 3 immunogenic neoantigen candidates from TIL cultures. **c,** n = 10 neoantigen candidates from n = 4 patients were analysed in total; n = 5 immunogenic neoantigen candidates from allogenic-matched PBMC cultures. HD, healthy donor; PBMCs, peripheral blood mononuclear cells; SFU, spot forming units; TIL, tumor-infiltration lymphocytes. (Tretter et al., 2023)

All 21 immunogenic neoantigen candidates were detected from RNA sources. Among them, 20 originated exclusively from RNA variants, while only one reactive neoantigen candidate was additionally identified from a somatic DNA variant (Figure 26 a). The variant effect and transcript type distribution of these variants was highly comparable to the distribution observed for all neoantigen candidate variants (Figure 26 b). In accordance with the findings for RNA-only variants

(Figure 16 b and c) and neoantigen candidates (Figure 23 e and f), also the majority of immunogenic neoantigen candidates from RNA variants harbored a detectable canonical sequence at the DNA level (Figure 26 c) and a substantial proportion of those were reported as A to G variants (Figure 26 d).

When examining the predicted binding affinities for all identified immunogenic neoantigen candidates with NetMHC4.0 (Andreatta & Nielsen, 2016) and MHCFlurry (O'Donnell et al., 2018) (data provided in Appendix 6.3), only 65% were defined as binders by at least one algorithm (defined as percentile rank <2% or predicted binding affinity <500nM). This indicates that one-third of immunogenic neoantigen candidates would have been missed if solely relying on binding prediction algorithms for the identification of neoantigens (see 3.5.2).

Overall, immunogenicity of neoantigens was observed across various tumor entities in different patients, including carcinoma, sarcoma, and melanoma (Figure 26 e, Appendix 6.3), suggesting that the identification of immunogenic neoantigen candidates is not limited to specific types of tumors.

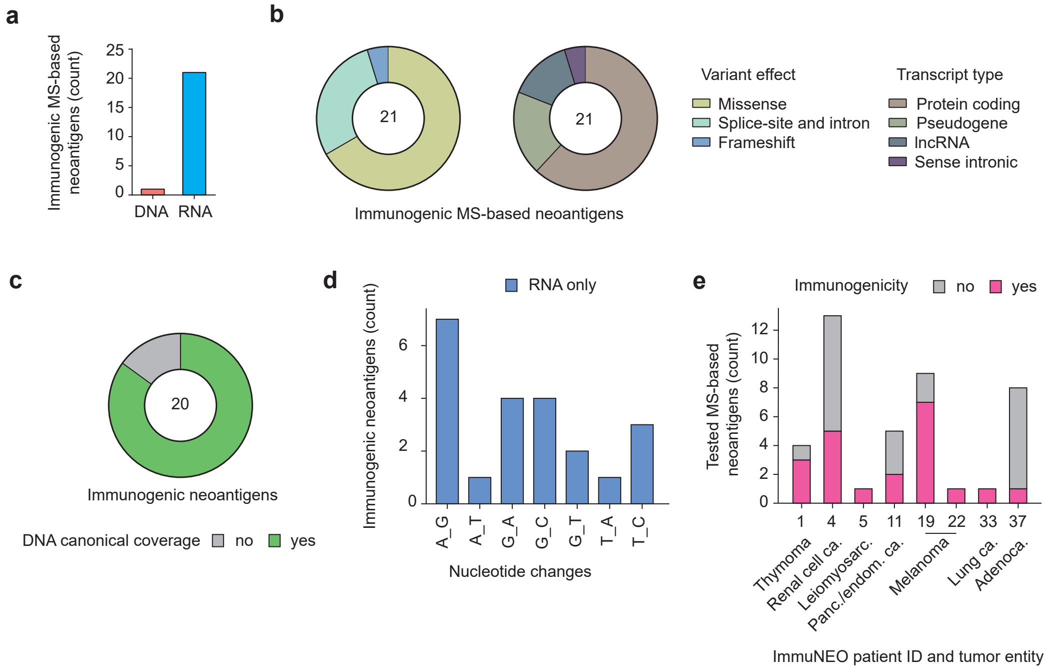

Figure 26: Characteristics of neoantigen candidates showing in vitro immunogenicity.
a, Genetic origin (DNA or RNA data) of the variants from which the immunogenic neoantigens were derived. **b,** Proportion of each variant effect type (left) and transcript biotype (right) of all variants producing immunogenic neoantigens. **c,** Pie chart illustrating the proportion of immunogenic neoantigens exclusively detected from RNA sequencing (RNA-seq) data, distinguishing cases where the respective canonical sequence was identified at the DNA level with a coverage of ≥ 3 reads (green) and where the respective region was not covered (grey, < 3 reads). **d,** Representation of the nucleotide exchange pattern of all single nucleotide variants generating immunogenic neoantigen candidates detected exclusively from RNA-seq data (n=22). **e,** Total number of tested neoantigen candidates for those patients showing immunogenicity, distinguishing between immunogenic (pink) and non-immunogenic (grey) ones. **a-b, e,** n = 79 neoantigen candidates from n = 24 patients were analysed in total; n = 8 patients harboured n = 24 immunogenic neoantigens. **c-d,** n = 23 neoantigen candidates from RNA variants. ca., carcinoma; endom., endometrium; Panc., pancreas. (Tretter et al., 2023)

In summary, immunogenic neoantigens were identified in 25% of all patients in this pan-cancer cohort, regardless of their tumor entity, when using a proteogenomic pipeline incorporating RNA transcriptomics of tumor samples for the identification of genetic variants. This data furthermore suggests that these RNA variants arise from aberrations in non-coding regions or RNA editing events, potentially contributing to the generation of immunogenic neoantigens.

3.6.2 Isolation of neoantigen reactive T cells from autologous PBMCs and TILs

With the aim of identifying neoantigen-specific T cell clones and TCRs as potential immunotherapeutic, 11 neoantigen-reactive bulk T cell cultures determined by acDC and ELISpot analysis were split into single cell cultures with 0.5 – 5 cells per well. In total 4,140 clones were seeded from which 361 expanded and were tested for neoantigen-reactivity in target cell co-cultures with subsequent IFN-γ ELISA as readout. From all of these 361 expanded clones only one clone reactive to neoantigen IN_22_A showed some slight response (Figure 27 a), however the reactivity was lost upon further cultivation and re-testing (Figure 27 b) and no neoantigen-specific clone could be isolated from autologous PBMCs or TILs.

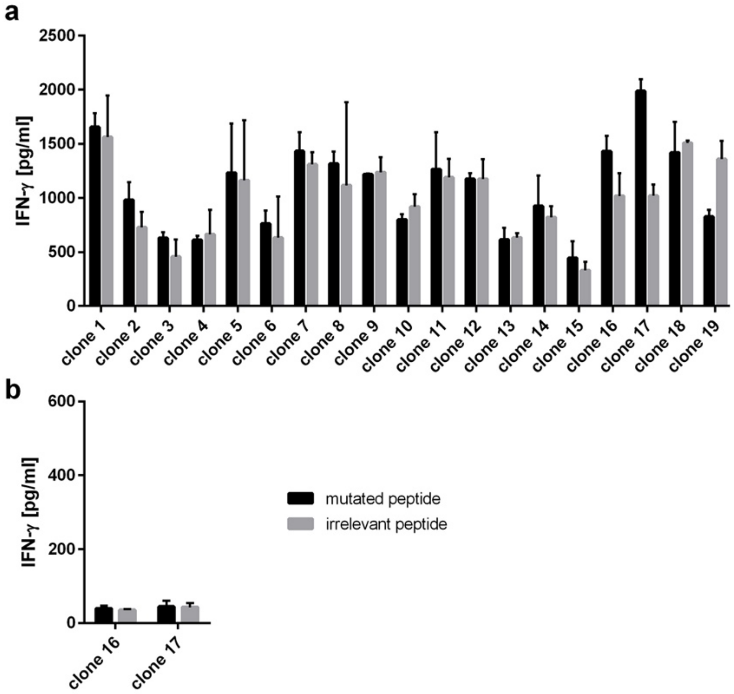

Figure 27: Reactivity assessment of neoantigen-specific T cell clones.
a, Representative reactivity assay for clonal selection of potentially neoantigen-specific single T cell clones. IN_22_A neoantigen-reactive bulk T cells from ImmuNEO-22 TILs (Figure 25 a) were seeded as single cells and expanded together with 50,000 irradiated feeder PBCMs in TCM supplemented with 5 ng/ml IL-7 and IL-15, 30 U/ml IL-2 and 30 ng/ml OKT-3. After 12 days, expanding clones were selected and half of the T cells were used for co-culture assays using neoantigen- and control peptide-pulsed HLA-matched target cells (autologous LCL) for reactivity assessment. After 24h of co-culture, the supernatant was taken off and analysed using IFN-γ ELISA assay. Shown are the concentration of IFN-γ in the supernatant for each condition as a mean of two replicates with standard deviation annotated. **b,** IFN-γ ELISA results of reactivity assays of re-expanded clones 16 and 17 from IN_22_A reactive ImmuNEO-22 TILs after 17 days of expansion. Shown are the concentration of IFN-γ in the supernatant for each condition as a mean of two replicates with standard deviation annotated. IFN-γ, Interferon-γ.

3.7 Multi-factor validation of neoantigen candidates

As described in 2.2.5, relaxed criteria for neoantigen candidate identification from MS data with the proteogenomics pipeline were used to increase the likelihood for detection of such low abundance targets. Setting the FDR to 5% on the peptidome level and using unfiltered variants on the genomic/transcriptomic level, however, can result in the false identification of non-canonical peptides and thus potentially unusable or unsafe targets for therapy purposes. Therefore, an in-depth validation pipeline for the verification and evaluation of the peptides and the DNA/RNA variants was developed and used to classify the identified neoantigen candidates. The following data was already published in Tretter *et al.*.

3.7.1 Peptide verification

On the peptidome level, the correctness of the identified peptide sequences of all neoantigen candidates was verified using two different approaches. Therefore, the experimental MS spectra of each neoantigen candidate was compared to the MS spectra of its cognate synthetic peptide and a predicted spectrum generated by Prosit. Neoantigen candidates with a normalized spectral contrast angle (SA) (Toprak et al., 2014) of at least 0.7 with either the synthetic or Prosit-predicted spectra was seen as matches (D. Wang et al., 2019). Out of 88 tested peptides, 41 could be verified using these criteria (47%, Figure 7, Appendix 6.3) while 19 candidates were close to the SA cutoff (22%). These candidates may still represent valid peptide-spectrum matches, although additional confirmation steps might be advisable to ensure their accuracy and reliability. Neoantigen candidates where the SA is below 0.5 (n = 28, 32%) cannot be verified as here the identified peptide sequence may not be correct and no proof for tumor-presentation of these peptides can be assumed. Applying stricter FDR filters for the MS data could lead to the reduction in discovery of such probably false positive neoantigen candidates, as seen in Figure 28, where FDRs below 1% result in less peptides with a SA value below the 0.7 cutoff. However, when comparing FDRs for all neoantigen candidates to the calculated SA, it can be observed that still many peptides above the SA cutoff can be found with “higher” FDRs between 1 and 5% (Figure 28, red rectangle). These neoantigen candidates would have been missed with stricter criteria in the peptidomic pipeline.

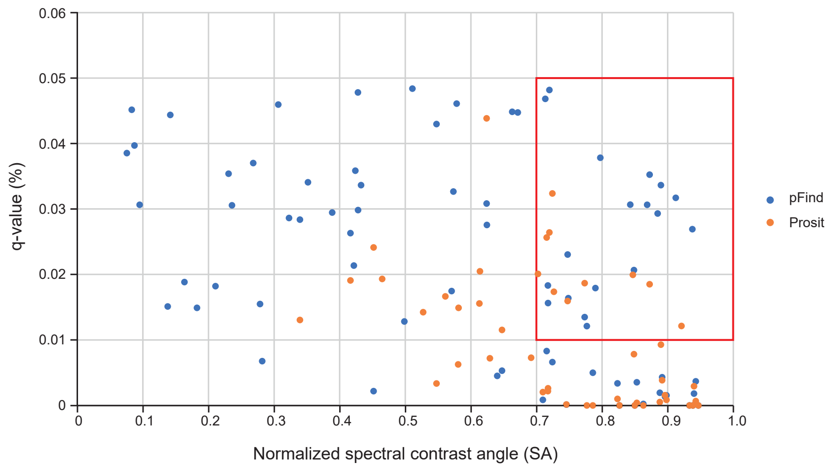

Figure 28: Verification of the applied false discovery rate during MS analysis.
The false discovery rate (FDR, depicted as q-values, FDR = q-value*100 in %) of 88 neoantigen candidate peptides with which they were found in the tumor by MS analysis is plotted against the corresponding best normalized spectral contrast angle (SA) between the measured and the synthetic or predicted spectra. Neoantigen candidates identified using pFIND are represented in orange, while those identified using Prosit are indicated in blue. FDR, false discovery rate; SA, spectral contrast angle. (Tretter et al., 2023)

As a second peptide verification, the experimental retention times (RT) of the liquid chromatography of all peptides were compared with predicted RTs calculated by Prosit. The majority of the experimental RT of the neoantigen candidates matched with the predicted RT (n = 45 candidates, green dots, absolute error of less than ± 8.56 min) as shown in Figure 29. For some peptides the RTs could not be accurately predicted by Prosit (n = 17, yellow dots) and were therefore not seen as verified nor non-verified.

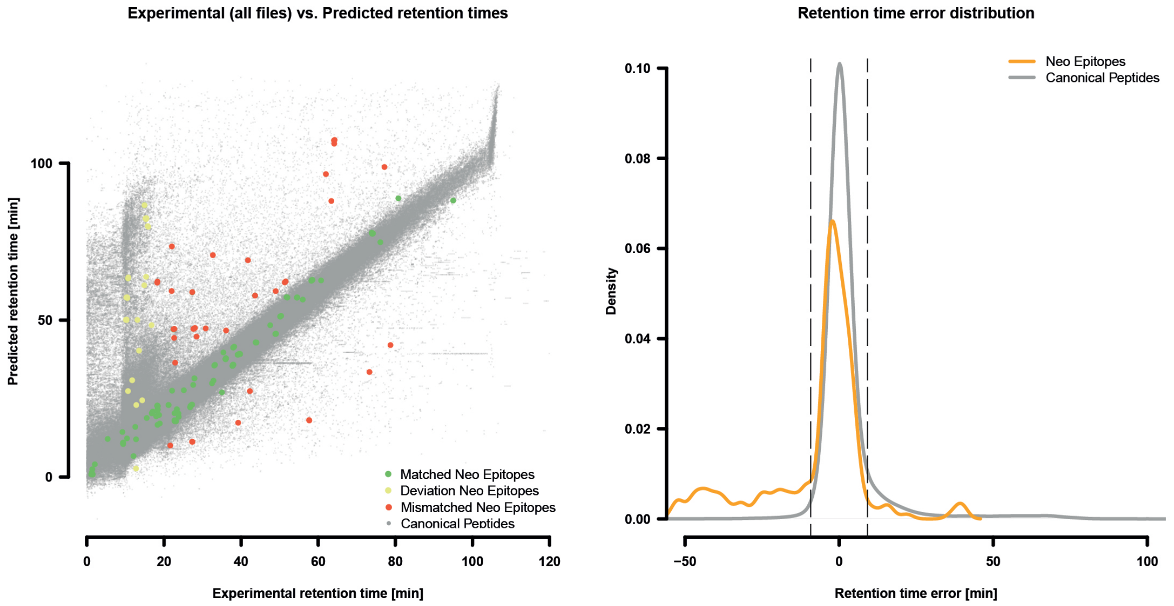

Figure 29: Peptide verification using predicted retention times.
Prosit was utilized to predict retention times (RT) for all identified canonical peptides and 88 neoantigen candidates. Predicted RTs were compared to the respective experimental RTs (left graph) and the error between both values was assessed (right graph). The RT error of all comparisons is plotted in the right graph (neoantigen candidates – orange line, canonical peptides – grey line) and the extreme of the upper whisker of the absolute error for all data points (within +/- 8.56 min, black dashed lines) was set as the threshold for verification. All neoantigen candidates (n = 88) were classified as matches (green dots in left graph) and mismatch (red dots in left graph) according to this threshold. For some peptides no correct RT could be predicted and values were excluded for verification (yellow dots in left graph). (Tretter et al., 2023)

3.7.2 Tumor-specificity assessment of RNA variants

As an additional validation step, the tumor-specificity of all neoantigen candidate variants was assessed, with a particular focus on RNA variants where no normal control was available. Given that RNA editing is a physiological process playing a role in healthy tissues, an evaluation of the presence of all 90 neoantigen candidate variants in normal tissues was conducted. This analysis involved examining more then 10,000 RNA-seq samples from 30 different healthy tissues accessed from the Genotype-Tissue Expression (GTEx, (Lonsdale et al., 2013)) project (Figure 30 a, Appendix 6.3). Out of the 90 candidates, 38 were completely absent from the GTEx healthy tissue RNA-seq samples and might therefore be seen as likely tumor specific. The other 52 neoantigen candidate variants either showed high prevalence (n = 16; found in more than 5% of samples), intermediate prevalence (n = 6; found in 1-5% of samples), low prevalence (n = 12, found in 0.1-1% of samples) or very low prevalence (n = 7; found in less than 0.1% of samples) in healthy tissues (Figure 30 a). For variants where the locus was not covered sufficiently (in less than 5% of all samples with at least 3 reads) no clear statement can be made and these variants were defined as not available (N/A, n = 11).

To account for these variants that were not covered sufficiently in the GTEx data set and to screen for rare patient-specific variants, total RNA-seq data from sorted CD8$^+$ TILs of ImmuNEO patients (Figure 13) was analyzed for the presence of the neoantigen candidate variants. In the CD8$^+$ TILs of these 8 patients indeed 8 neoantigen candidate variants were found, however only one of them (variant of IN_19_F) was present in the patient it was originally identified in (Figure 30 b). Importantly, this variant was additionally identified with high prevalence in the GTEx data and thus did not meet he validation criteria in the first place. Two neoantigen candidate variants (variant of IN_4_B and IN_13_A) were not detected in the GTEx data with sufficient coverage but were found in the CD8$^+$ TIL RNA-seq analysis (Figure 30 b).

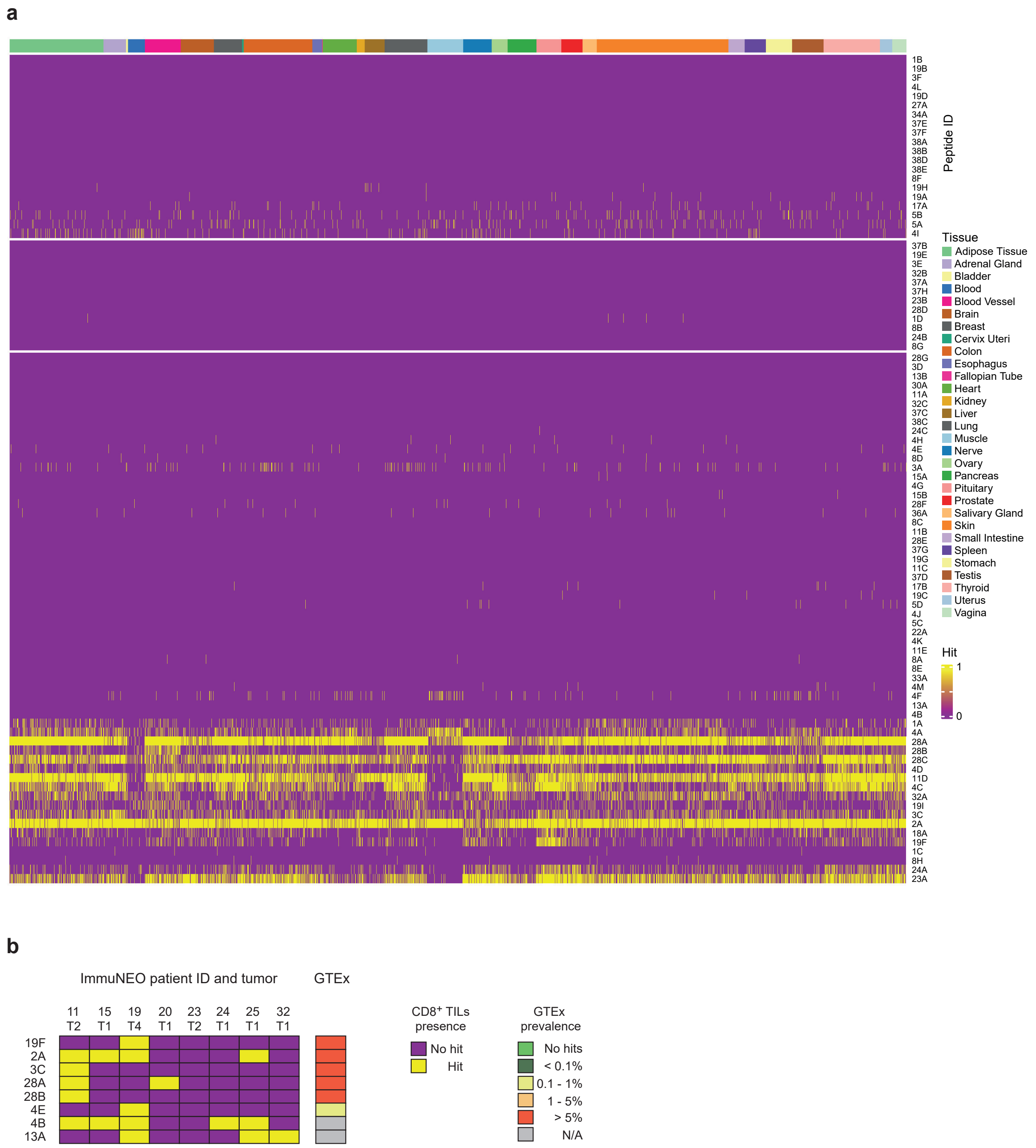

Figure 30: Prevalence of variants inducing neoantigen candidate in healthy tissue.

a, Heatmap depicting the prevalence (hit = min. 1 read, yellow) of each neoantigen candidate variant in RNA expression data of 10,269 samples from 30 different healthy tissue accessed from the GTEx data base (Lonsdale et al., 2013). The respective healthy tissue type is annotated. **b,** Heatmap displaying the prevalence of neoantigen candidate variants (left annotation) within bulk RNA sequencing data of sorted CD8$^+$ T cells from ImmuNEO patients (upper annotation). The prevalence of each variant found in the GTEx data set (see a) is annotated on the right. Data points were considered not applicable (N/A) if the locus lacked sufficient coverage (less than 3 canonical reads in less than 5% of samples). GTEx, genotype-tissue expression; N/A, not applicable; T, tumor; TILs, tumor-infiltrating lymphocytes. (Tretter et al., 2023)

As variants don’t have to necessarily be tumor specific but could also be tumor associated when their frequencies are increased in comparison to normal tissue, the variant frequencies found within the GTEx healthy tissue samples and the frequency found within the tumor were compared (Figure 31). Only few candidates (IN_2_A, IN_19_A, IN_19_F, IN_28_B) displayed potential tumor-associated RNA-overediting, while for the majority of variants no elevate frequencies were detected compared to normal tissues.

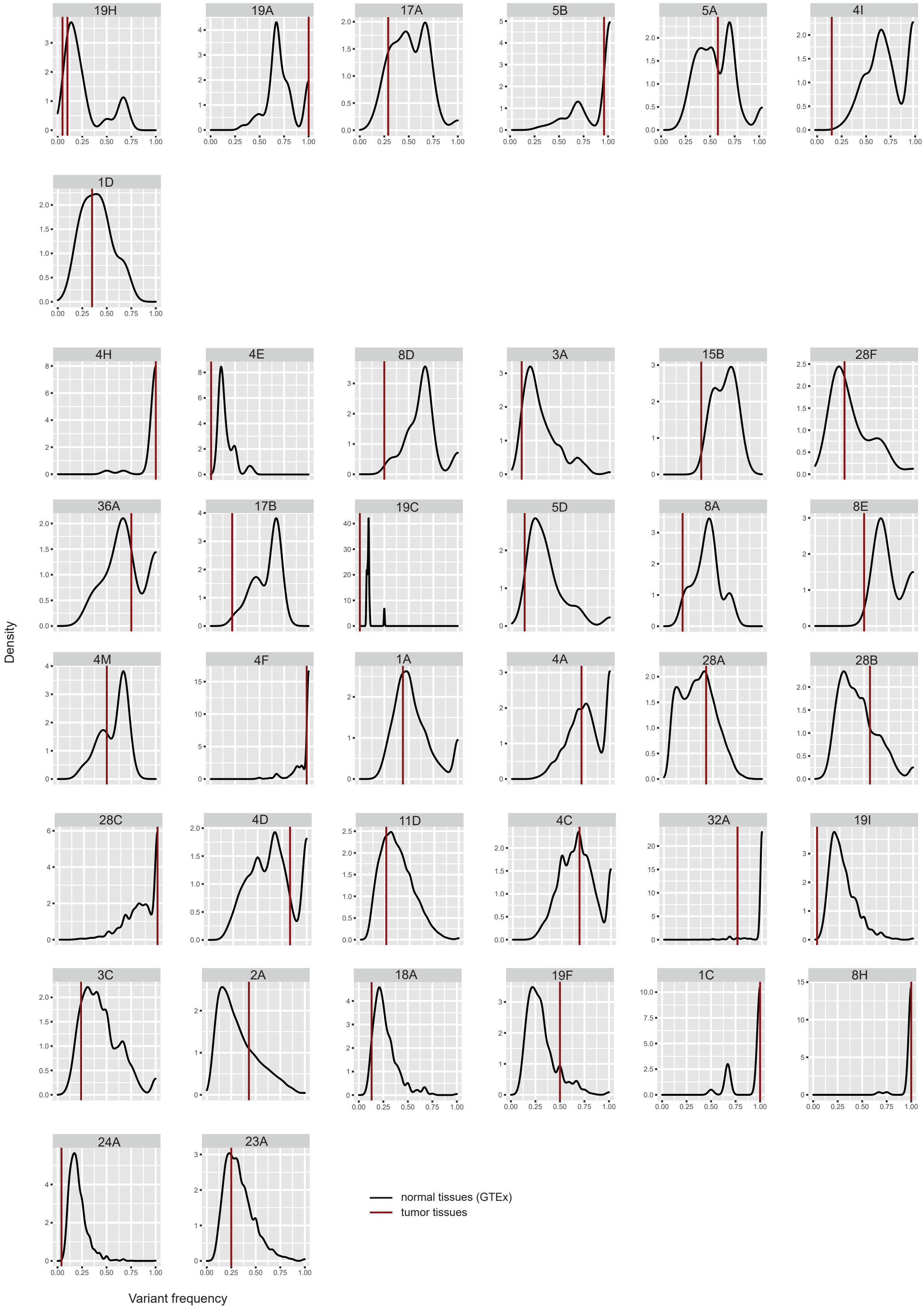

Figure 31: Distribution of variant frequency for all neoantigen candidate variants in tumor and heathy tissue.
The variant frequency for each neoantigen candidate variant was plotted across all analyzed samples in the GTEx dataset (10,269 samples from 30 different tissues) (Lonsdale et al., 2013). The variant frequency observed within the patient's tumor sample is indictaed by a red line for reference. GTEx, genotype-tissue expression. (Tretter et al., 2023)

3.7.3 Neoantigen validation and prioritization

For a final neoantigen candidate validation and potential prioritization of peptides, all above-described validation factors were incorporated. For neoantigen verification, the SA and RT were combined and complete matches for both were seen as completely verified. Neoantigen candidates, where the SA met the validation criteria (above 0.7) but the RT could not be predicted by Prosit, were seen as partly verified, whereas neoantigen candidates not meeting any of the quality criteria were seen as not verified. In addition to the neoantigen candidate verification, also the GTEx prevalence was set as a validation criterion. Variants with prevalence of up to 5% in healthy tissue were still seen as non-canonical and thus potentially promising targets.

When combining all validation criteria, the neoantigen candidates were categorized into three groups: 20 highly promising candidates (Figure 32, top), that have successfully passed multiple validation criteria and exhibit strong potential for further investigation. 12 potentially promising candidates (Figure 32, middle), that require further verification of their peptide sequence or prevalence in normal tissues. And 59 not very promising candidates (Figure 32, bottom), that either lack robust proteomic verification or are commonly detected in normal tissues. These 59 neoantigen candidates were excluded, while the remaining 32 highly and potentially promising candidates were considered as validated (detailed information on each peptide summarized in Appendix 6.3).

Applying the validation criteria, an enrichment for neoantigen peptide ligands found by both algorithms, pFind and Prosit, and for nonameric peptide ligands from protein coding transcripts was observed (Figure 33 a and c, g). Still, validated neoantigen candidates were found in nearly 50% of patients (15 of 32), originating predominantly from RNA variants and showing similar characteristics as the total pool of peptides and neoantigen candidates (Figure 33 b and d-g)

Interestingly, out of these 32 highly and potentially promising candidates, 8 elicited an immune response. However, also reactivities were found towards some peptides not meeting the validation criteria. When looking at the binding prediction using NetMHC and MHCFlurry, 90% of all highly promising neoantigen candidates were also predicted as weak or strong binders by at least one algorithm. In contrast, only 58% and 51% of the potentially promising and excluded candidates were predicted as binders, respectively.

In summary, adding a post-analysis validation step enriched the pool of identified neoantigen candidates for more clinically promising targets that might have been missed when applying more strict criteria from the beginning.

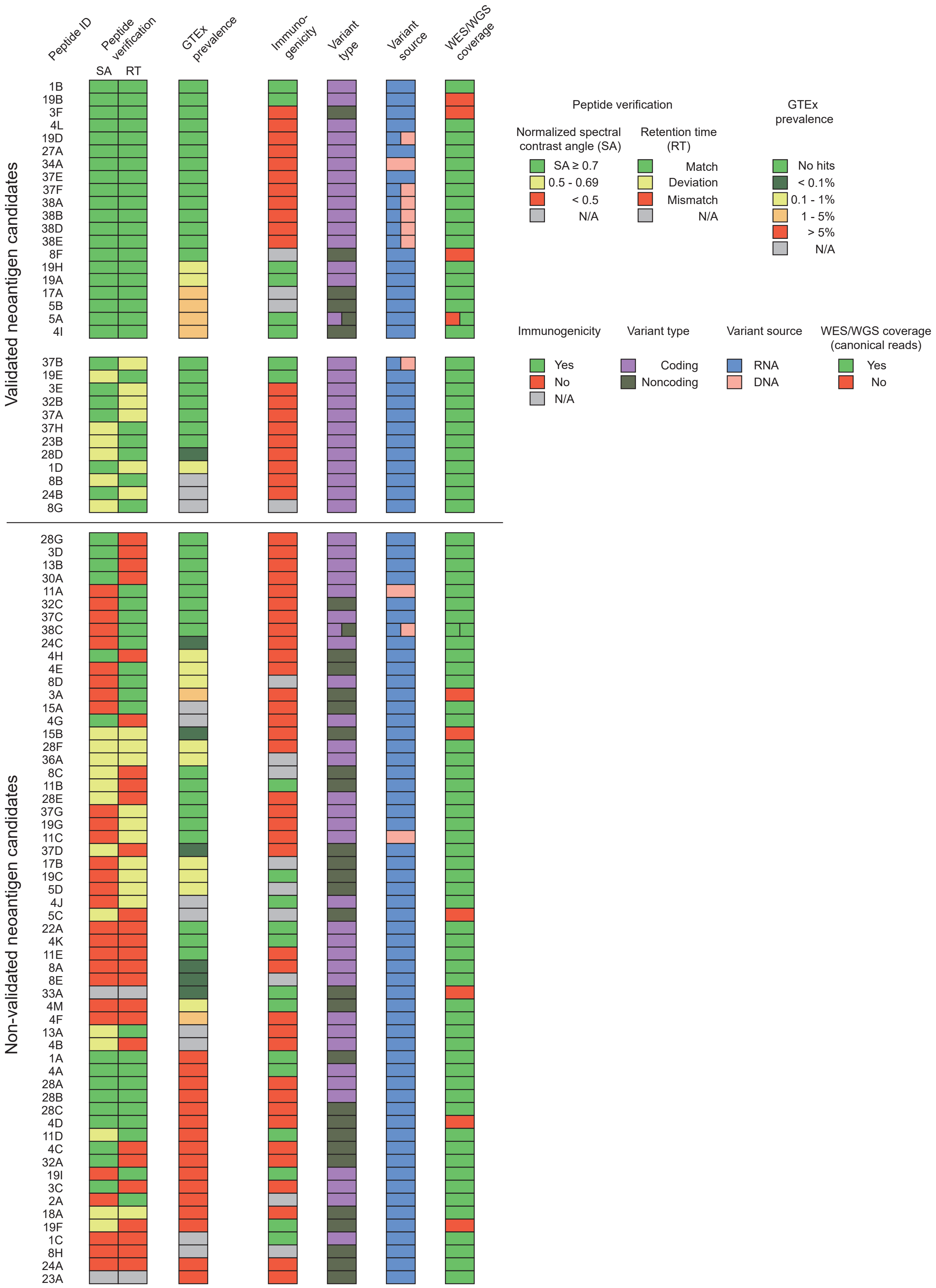

Figure 32: In-depth validation of neoantigen candidates.

Two validation steps were applied to the neoantigen candidates. Initially, the MS-measured peptides were verified by comparing their spectra with the spectra of measured respective synthetic peptides spectra or spectra predicted using Prosit. The best normalized spectral contrast angle (SA) from both methods was utilized, categorizing the neoantigen candidates as matches (green, SA ≥ 0.7, n = 41), likely matches (yellow, SA 0.5-0.69, n = 19) and mismatches (red, SA < 0.5, n = 28). Additionally, the retention time (RT) of each peptide was predicted using Prosit and compared to the MS-measured RTs. Matches were identified when RT errors between predicted and measured RTs were less than ± 8.56 (green, see Figure 29, n = 45). Neoantigen candidates were considered validated when both SA and RT matched between experimental and synthetic/predicted values. In the second step, the tumor specificity of all variants was assessed by analyzing the prevalence of each variant in RNA expression data from healthy tissue samples (n = 10,269 samples from 30 different healthy tissue) from the GTEx data base (Lonsdale et al., 2013). A variant was considered present when a min. of 1 noncanonical read was found and the locus was covered with a min. of 3 canonical reads in a min. of 5% of analyzed samples. Variants with a prevalence ≤ 5% were interpreted as potentially tumor specific. Both validation criteria, including peptide verification and tumor-specificity, were combined, resulting in validated neoantigen candidates (promising candidates, top, n = 20 and potentially promising candidates, middle, n = 12; total n = 32) and non-validated neoantigen candidates (bottom, n = 58) separated by a line. Furthermore other relevant parameters are annotated in the graph, including the in vitro immunogenicity (Figure 27) of all tested peptide, the type and source of each neoantigen candidate variant and the coverage on DNA level within the tumor (min. 3 canonical reads). RT, retention time; SA, spectral contrast angle, WES, whole exome sequencing; WGS, whole genome sequencing. (Tretter et al., 2023)

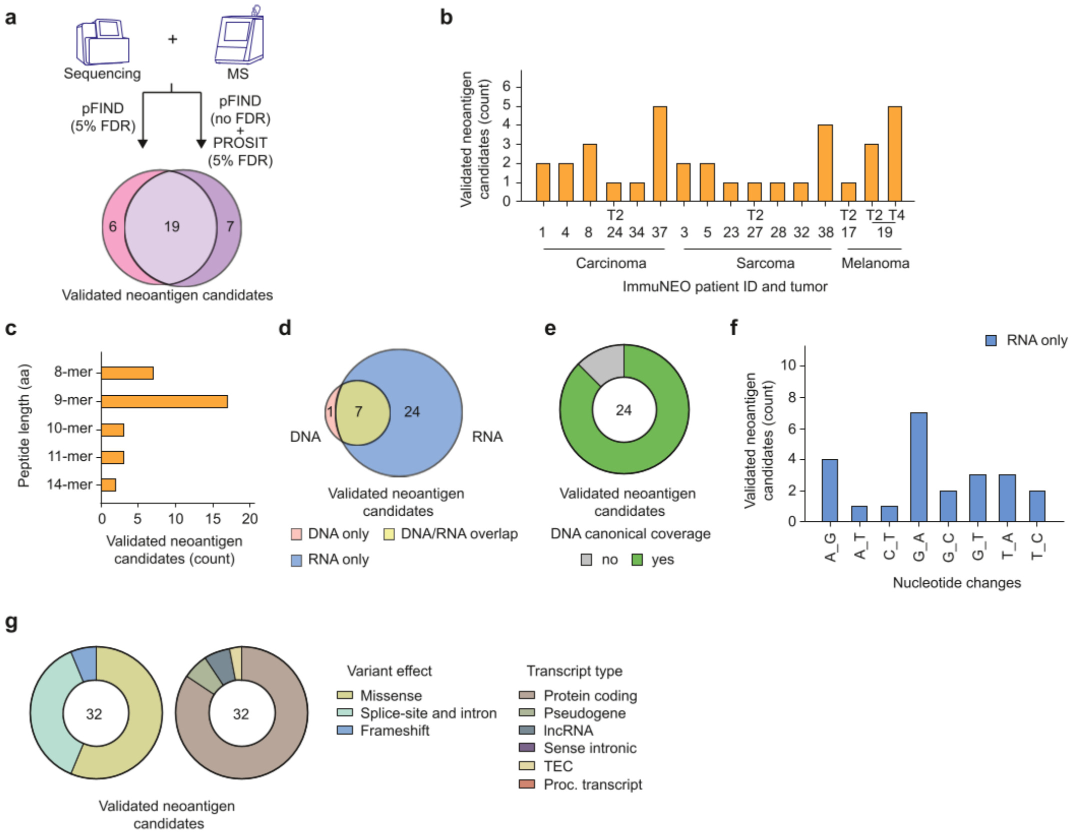

Figure 33: Characteristics of validated neoantigen candidates.

a, Number of validated neoantigen candidates based on the bioinformatics tool used for their identification. pFIND (v3.1.5) (Chi et al., 2018) was utilized at a false discovery rate (FDR) of 5% on the spectral level for identifying non-canonical 8-15mer neoantigen candidates. The machine learning tool Prosit (Gessulat et al., 2019) was additionally integrated for neoantigen identification at 5% FDR, using unfiltered pFIND data to rescore the peptide spectra matching to the patient-specific ORF database. Neoantigen candidates were validated based on their tumor-specificity and a peptide verification. A total of n = 39 tumor samples from n = 32 patients were analysed and n = 27 tumor samples from n = 24 patients harboured n = 90 neoantigen candidates of which n = 16 tumor samples from n = 15 patients harboured n = 32 validated neoantigen candidates. **b,** Number of validated neoantigen candidates per tumor sample, sorted by tumor entity. **c,** Length distribution in amino acids (aa) of all validated neoantigen candidates. **d,** Source (DNA or RNA data) of the variants from which the validated neoantigen candidates were derived. **e,** Proportion of validated neoantigen candidates identified exclusively from RNA sequencing (RNA-seq) data, distinguishing cases where the respective canonical sequence was identified at the DNA level with a coverage of ≥ 3 reads (green) and where the respective region was not covered (grey, < 3 reads). **f,** Representation of the nucleotide exchange pattern all single nucleotide variants generating validated neoantigen candidates detected exclusively from RNA-seq data. **g,** Proportions of each variant effect type (left) and transcript biotype (right) of all variants inducing validated neoantigen candidates. **a-g,** n = 39 tumor samples from n = 32 patients were analysed in total; n = 16 tumor samples from n = 15 patients harboured n = 32 validated neoantigen candidates; n = 1 validated neoantigen candidates from DNA variants; n = 7 validated neoantigen candidates from DNA and RNA variants; n = 24 validated neoantigen candidates from RNA variants. aa, amino acids; MS, mass spectrometry; Proc., processed; T, tumor; TEC, to be experimentally confirmed. (Tretter et al., 2023)

3.8 Integration of multi-omics data sets

3.8.1 Correlation of neoantigens with phenotypic features

Finally, the aim of this study was to identify potential multi-omics and tumor microenvironmental factors that affect or can be related to neoantigen load and also the potential immunogenicity of these neoantigens in order to better stratify patients for potential T-cell based immunotherapies. The resulting data was already published in Tretter *et al.*.

Therefore, Spearman`s rank correlation test of the number of validated and non-validated MS-based neoantigens (total numbers and immunogenic peptides) with the immunophenotyping (3.2.1) and immunopeptidomic data (3.4.1) was performed. Since all neoantigen candidates were identified from the MS spectral data, there was a strong correlation between both the total number and the number of validated immunogenic neoantigens with the size of the MS-immunopeptidome (Figure 34 a). Additionally, the total number of validated immunogenic neoantigen candidates correlated significantly with several features of the TME, such as the total frequency of CD3$^+$ T cells, CD8$^+$ T cells and CD8$^+$ Teff cells (Figure 34 a). Importantly, a generally more exhausted phenotype of the CD8$^+$ T cells and especially the Tem subset (Figure 34 a) was significantly associated with a higher number of immunogenic validated neoantigens identified within the patients' tumors.

When grouping all patients according to the presence or absence of immunogenic validated neoantigen candidates and comparing the above-described features by Wilcoxon rank-sum test between these groups, most of the observed significant correlations become even more eminent (Figure 34 b), although the cohort size is rather small.

In summary, the findings indicate that the existence of immunogenic validated neoantigen candidates within this cohort is associated with a more immunologically active TME and a high T-cell infiltration.

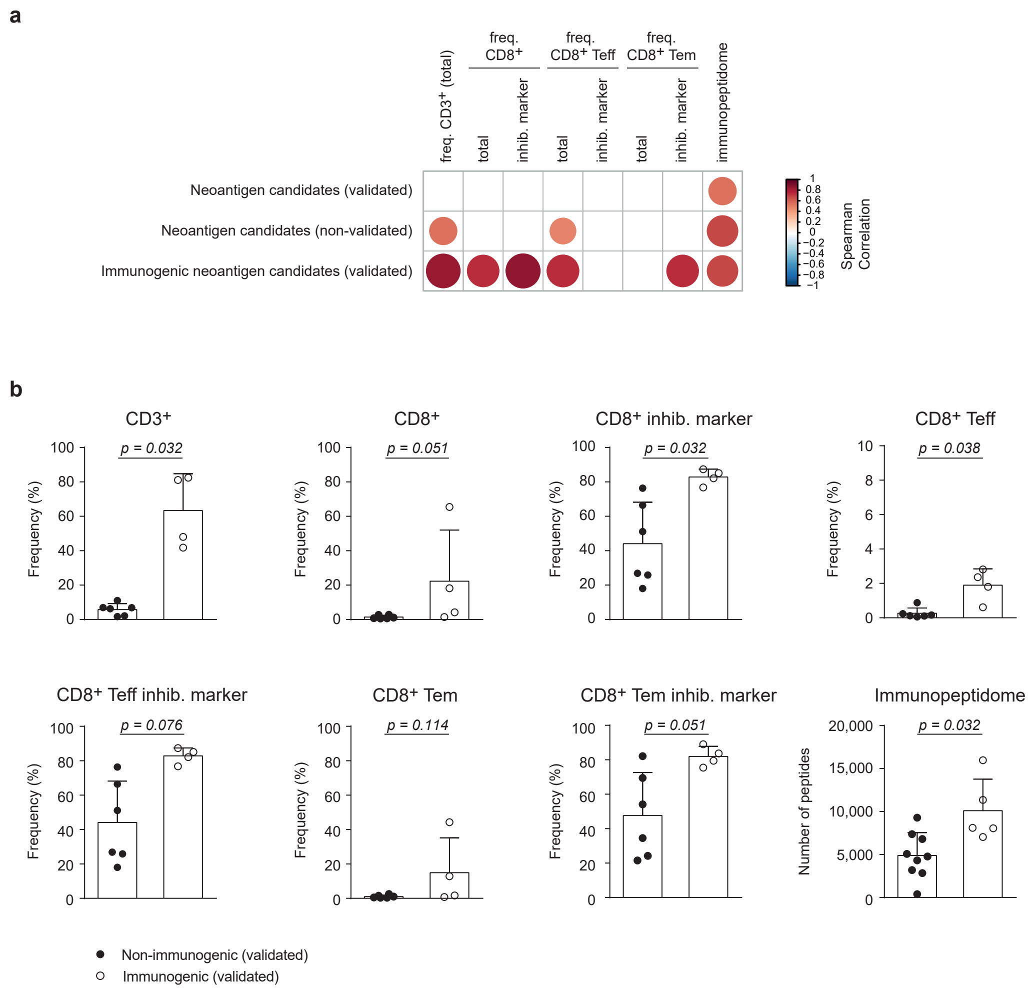

Figure 34: Correlation of neoantigen load and immunogenicity with tumor microenvironment characteristics and multi-omics features.

a, Correlation matrix visualizing significant Spearman correlations ($p \leq 0.05$) between various phenotypic parameters as well as the immunopeptidome size with the number of validated and non-validated neoantigen candidates as well as immunogenic validated neoantigen candidates. The Spearman correlation coefficient (Rho) is displayed in color and size of dots. Statistical analysis was performed using a single representative tumor sample per patient. **b,** Bar graphs comparing the frequencies of several immune cell subsets and the immunopeptidome size of patients with and without immunogenic validated neoantigen candidates. Statistical differences between groups were tested using Mann-Whitney U test with Benjamini-Hochberg procedure for correction for multiple testing. Respective p values are annotated within the graph for each test. Statistical analysis was performed using a single representative tumor sample per patient. **a,** Phenotypic correlations with validated (n = 19), non-validated (n = 19) and immunogenic validated (n = 10) neoantigen candidates. Immunopeptidome correlations with validated (n = 32), non-validated (n = 32) and immunogenic validated (n = 14) neoantigen candidates. **b,** Phenotypic correlation with immunogenic (n = 4) and non-immunogenic (n = 6) validated neoantigen candidates. Immunopeptidome correlation with immunogenic (n = 5) and non-immunogenic (n = 9) validated neoantigen candidates. Freq., frequency; inhib., inhibitory; Teff, T effector cells; Tem, T effector memory cells. (Tretter et al., 2023)

3.8.2 Correlation of multi-omics features with patients' response to immunotherapy

A possible influence of each parameter to the response to ICI was of particular interest as well. Therefore, the subset of patients who received ICI prior to and after tumor resection (or both) were grouped into non-responder and responder (mixed response and good response combined). Using receiver operating characteristic (ROC) curve evaluation and Wilcoxon-Mann-Whitney U-test, a beneficial influence of a higher total number of genetic alterations (AUC=0.83, U-test p=0.082, Figure 35 a), although not significant in this small cohort. Several other parameters show non-significant trends (data not shown), for example a bigger over all lymphocyte infiltration (frequency CD3$^+$ cells, AUC=1, U-test p=0.1) as well as other immunological features such as higher infiltration of CD8 Tn cells without any expressed marker (no inhib. markers AUC=1 p=0.1; no activ. markers AUC =1 p=0.077), a higher frequency of CD8 Tem in general (AUC=1 p=0.1) and a higher amount of Tem expressing a least one inhibitory marker (AUC=1 p=0.1) show a potential beneficial influence on the ICI response of patients, although only 6 samples were included into this analysis.

When looking at the response type, where patients receiving ICI were stratified into no, mixed and good responders, the identified non-significant correlation of the response to ICI to the total number of mutations, and the number of MS-neoantigens was defined in more detail: Patients already showing a mixed response to ICI seem to have a higher number of total mutations in comparison to patients showing no response (Figure 35 b). Furthermore, stratifying mixed and good responders, a significant correlation between a larger wt peptidome (1% FDR) and a good response in comparison to no response to ICI was observed (p=0.036, Figure 35 c).

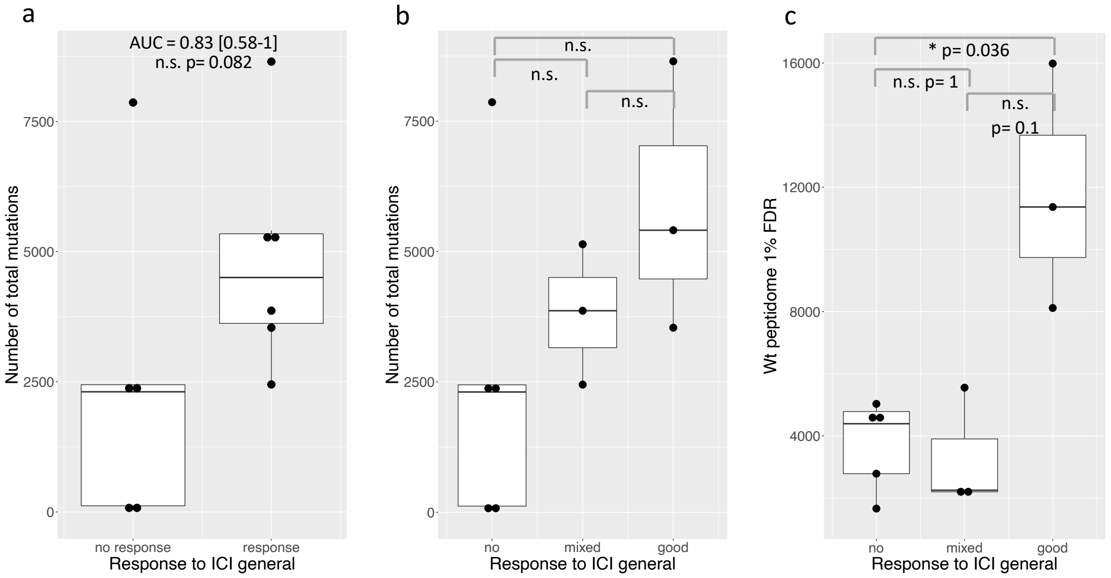

Figure 35: Correlation of genomic and peptidomic features with the response to immune checkpoint blockade.
a, Correlation analysis of the response to immune checkpoint inhibition (ICI) to several experimental parameters by Wilcoxon rank-sum test (U-test) and receiver operating characteristic (ROC) curve. Correlation to the number of total mutations is shown as box plot. P-values of U-test as well as the area under the curve (AUC) of the ROC curve with confidence intervals in brackets are shown. **b,c,** Response types to ICI divided into no response, mixed response and good response were correlated to several experimental parameters using Kruskal-Wallis rank sum test (H-test) for overall correlations and subsequent U-test for pair-wise correlations. Correlations to the number of total mutations (**b**) and the size of the immunopeptidome (**c**) are shown in box plots. P-values for pair-wise correlations are annotated. Statistical analysis was performed using a single representative tumor sample per patient. Significant correlations are labeled with * p ≤ 0.05, ** p ≤ 0.01, *** p ≤ 0.001. **a-c,** n = 11 patients. AUC, area under the curve; FDR, false discovery rate; ICI, immune checkpoint inhibition; MS, mass spectrometry; n.s., not significant.

Of note, factors such as the patients age and the tumor cell content did not influence survival or response to ICI. Also, the number of peptides identified at 1% FDR as well as 5% FDR did not correlate directly to the tumor weight used for MS, although a higher HLA expression is slightly indicative for a bigger peptidome per gram tumor (data not shown).

4. Discussion

Systemic pan cancer studies and also therapy approaches for personalized cancer therapies have gained significant importance within the past years and have changed the way of cancer treatment. In addition, especially immunotherapy has shown great potential (Eggermont et al., 2016; Topalian et al., 2019), also in a tissue agnostic manner.

Clinical application of immunotherapies such as mRNA-based vaccines (Sahin et al., 2017) and cellular immunotherapy (Tran et al., 2016) have advanced tremendously over the past years. Yet, the identification of tumor-specific and therapeutically relevant targets for such therapies, mainly focusing on neoantigens, is still critical. In the past, research in this field has predominantly focused on cancer genomics and *in silico* epitope prediction models to identify potential neoantigens (Verdegaal et al., 2016). However, there is considerable potential for significant advancements by incorporating alternative approaches, such as proteogenomics, as demonstrated by other research groups (Chong et al., 2020; Laumont et al., 2018) and us (Bassani-Sternberg et al., 2016; Wilhelm et al., 2021).

The findings presented in this thesis underscore the significance of RNA as a crucial source for identifying neoantigens and shared tumor antigens, achieved through an enhanced proteogenomic pipeline in a thoroughly characterized pan-cancer cohort. Furthermore, a comprehensive validation analysis was added to the identification pipeline that could be used to guide selection of promising neoantigen candidates for clinical application. Integrating proteogenomics with phenotypic and functional analyses furthermore enabled the association of identified neoantigen candidates with immunological features, validating their potential to elicit T cell-driven immune responses. In the following, the presented results will be discussed in more detail, highlighting potential limitations but also discussing future applications of the gained knowledge.

4.1The cohort – small but representative

Despite the relatively small cohort size and the high diversity in tumor entity, disease stage, treatment history, age, and gender, this study successfully confirmed prognostically significant biomarkers already established for various malignancies.

The significant positive correlation between patients` survival and the number of somatic mutations seen within this cohort validates the TMB as a prognostic biomarker, consistent with previous findings across various cancer types and selected cross-entity studies (Litchfield et al., 2021; Rizvi et al., 2015; Samstein et al., 2019; Snyder et al., 2014). Additionally, elevated levels of CD8$^+$ T cells expressing inhibitory markers, indicative for a dysfunctional T cell state within the TME (Thommen & Schumacher, 2018), were associated with poor clinical outcomes, consistent with prior findings

(Zheng et al., 2021). While not statistically significant in this small cohort, the overall infiltration of CD8$^+$ T cells showed a positive correlation with favorable survival, aligning with findings from previous studies (Bruni et al., 2020; Galon et al., 2006; Leffers et al., 2008; Oshi et al., 2020). Furthermore, a positive influence of a higher mutational burden as on the response to ICI was observed within this pan cancer cohort, as already shown recently (Litchfield et al., 2021; Pender et al., 2021; Samstein et al., 2019).

The confirmation of these pre-defined biomarkers in such a small and heterogeneous cohort on the one hand indicates the representativeness of our patient group and on the other hand indicates that these biomarkers have a strong prognostic power.

Nevertheless, all statistical analyses within this cohort need to be taken with caution due to low sample size and high heterogeneity. Thus, the power of statistical testing is limited.

4.2 The immunopeptidome – potential shared TAAs

Within this study the HLA class I immunopeptidome could be defined for every sample analyzed with a very high quality. This was determined by looking at the size distribution of peptides ranging from 8-15mers with the majority of 9meric peptides, as expected for HLA class I molecules and previously described (Bassani-Sternberg et al., 2015, 2016). Furthermore, the motif deconvolution showed that nearly all identified peptides matched to the respective patient HLA class I alleles (mean of 95% over all patients) suggesting that real HLA-bound peptides were eluted and measured and only a mean of 5% of peptides could have been falsely identified.

The total size of the peptidome varied greatly between patients and even between different tumor samples of the same patient, however, no clear entity dependent patterns could be identified. Also, the correlation of the immunopeptidome size with neither the level of HLA expression nor the used tumor mass could be observed, suggesting that other factors may influence the size of the immunopeptidome. However, patients with bigger immunopeptidomes showed better responses to ICI, as the immune system might have more potential antigens to recognize in comparison to tumors presenting less peptides on their surface, although the picture is probably much more complex (Kraemer et al., 2023).

Interestingly, although this cohort is so heterogeneous, several shared peptides were observed between different patients and samples. These peptides originated from genes associated to cancer which are therefore more likely to be expressed, processed and presented than other genes/proteins (e.g. housekeeping genes etc.) and thus could represent potential TAAs. However, it would need to be determined in additional analyses and extensive experiments if these genes are actually (over-) expressed in this cohort. Also, tumor reactivity in general and of course also tumor specificity and cross-reactivity of each peptide would need to be investigated to determine the biological relevance and by that the potential of these shared peptides as TAAs. The largest group of 18 shared peptides were all predicted to bind to two very frequent HLA alleles, namely HLA-A0301 (14.74% in Europe (Gonzalez-Galarza et al., 2020b)) and HLA-A1101 (5.63% in Europe (Gonzalez-Galarza et al., 2020b)), and all patients where theses peptides were found also had those HLA alleles. These peptides could therefore also be interesting common TAAs for a larger population of patients having these HLA alleles.

Other potential TAAs originating from CTAs were identified within this study. CTAs have already been used as TAAs for the development of TCR-based but also vaccination-based immunotherapy approaches (see 1.2.1) and therefore represent interesting and important targets for therapy development. We could not only show that peptides from CTAs can be identified within this heterogeneous cohort but that they are also potentially entity agnostic as they were found in various tumor types. Mainly, peptides from ATAD2 and SPAG9 were found, that are both described as oncogenes in several cancers (Nayak et al., 2021; Pan et al., 2018). SPAG9 is a more recently identified CTA which showed expression and influence on tumorigenesis in several different cancer types and was found to be able to induce humoral immune responses against tumors (Pan et al., 2018). Recently, recombinant SPAG9 has been tested in the context of DC-based immunotherapy and showed also cellular immune responses by CD4^{+}, CD8^{+} T cell and NK cell activation in cervical cancer, however the data is still preliminary (Dhandapani et al., 2021). Furthermore, different peptides from PRAME, a CTA currently under investigation as target for immunotherapy in different cancers (NTC02743611, NCT03503968, NCT03686124, see

Table **3**), were found in this cohort.

Most interestingly, we also found shared CTA peptides (identical in their sequence) in several patients that represent interesting therapeutic targets. Here, also several peptides originated from ATAD2 were found and even one peptide from cTAGE5 was present in 12 patients. However, as previously emphasized, several further investigations would be needed to validate these shared potential TAAs and determine their therapeutic potential.

As not only TAAs but specifically TSAs in the form of neoantigens were of interest in this thesis, the further analysis focused on the neoantigen identification and validation pipeline.

4.3 Improvement of neoantigen identification – RNA with its potential but also limitation

To enhance the probability of neoantigen identification beyond our previously published proteogenomic strategy (Bassani-Sternberg et al., 2016), tumor RNA was incorporated as a supplementary source for variant detection. Additionally, the mass spectrometry analysis was improved by integrating additional peptide-spectra matching algorithms. The integration of RNA-seq data into the pipeline offers two key advantages. Firstly, RNA-seq complement WES in detecting somatic mutations, expanding the scope of discoveries, as demonstrated in glioblastoma multiforme (Coudray et al., 2018). Secondly, RNA-seq is able to identify variants not present at the DNA level but arising from RNA processing events such as alternative splicing or RNA editing (Saha et al., 2017; Tan et al., 2017). Previous reports have highlighted that RNA dysregulation and RNA editing events contribute to the diversification of the cancer proteome (Peng et al., 2018; Yang & Nam, 2020). In fact it was seen in this study that the number of total but also shared variants and potential neoantigens increased substantially by incorporating RNA-seq into the proteogenomic pipeline. The different improvements, their impact but also their limitations will be discussed in more detail in the following sections.

4.3.1 RNA sequencing – the potential of RNA variants

Variant detection using RNA-seq has already been employed in numerous studies for the identification of neoepitopes (Laumont et al., 2018; Merlotti et al., 2023; Zhou et al., 2020). Within this cohort a large number of variants were detected from RNA sequencing data only, which were found independent of the tumor entity and also did not correlate to the number of DNA variants found in the same patient. Thus, RNA variants might represent a rich and additional source for potential tumor-specific targets also for entities and patients with low TMB. However, no correlation of the number of RNA variants with the survival was observed, as seen for DNA variants, suggesting that the fraction of RNA variants influencing immunogenicity-associated responses and improved survival is relatively small compared to the sheer quantity.

By including RNA-seq for variant calling, we identified alterations in non-coding regions, such as pseudogenes, regulatory RNAs and regulatory regions that were shown to be non-canonically translated and can thus be sources of neoantigens (Chong et al., 2020; Laumont et al., 2018; Ouspenskaia et al., 2022; Ruiz Cuevas et al., 2021).

Nearly 60% of all RNA variants were found in such noncoding regions within this study in line with previous publications (Ruiz Cuevas et al., 2021). Alternatively, these kinds of variants could only be identified from DNA using WGS instead of WES, however this would be way more expensive and, at least to date, not affordable on a regular basis within the clinic.

The remaining 40% of RNA variants were found on protein coding regions and can in part be explained by intronic splice site variants that can only be detected by RNA-seq or WGS. Also, some of the variants in coding regions were not covered by DNA seq and thus these RNA variants could complement DNA sequencing to call somatic variants (Coudray et al., 2018). However, for a big proportion of RNA variants, a canonical sequence was detected on DNA level (see Figure 16b), indicating other mechanisms on RNA level leading to such variants. One potential explanation could be RNA editing events as described in 1.2.2 and indeed this data shows approx. 40% of variants with the editing-associated A-to-I (seen as A-to-G in sequencing) SNV pattern typical for ADAR-based editing. Also, "alternative mRNA editing" patterns such as C-to-U, U-to-C, and G-to-A have been observed, which may, at least in part, account for the remining RNA-only variants (Christofi & Zaravinos, 2019; B. R. Rosenberg et al., 2011). However, research in this field just started growing in the past years and further investigations would need to be performed to understand the nature and biology behind these variants.

As a much larger proportion of those RNA variants was shared between samples and patients than observed for DNA variants, RNA variants could be a rich source for interesting pan-cancer targets. Therefore, it is crucial to understand the tumor specificity or association of such variants before further application into the clinic.

The genomic and transcriptomic data in this study clearly shows that including RNA sequencing for variant calling can supplement somatic variant calling but most importantly broadens the spectrum of potential tumor-associated or -specific variants irrespective of the tumor entity. Not only for single patients with a low TMB but also for cross-entity approaches RNA variants present a great source for potential targets.

4.3.1.1 Limitations and perspectives

Nevertheless, RNA-seq has its inherent limitations, especially concerning variants originating from RNA processing events. Generally, the increased susceptibility to sequencing errors during reverse transcription poses one of the additional challenges. Furthermore, the validation of RNA variants is challenging as it cannot be fully achieved using matched-normal DNA samples. The use of matched-normal RNA samples as controls is constrained as well, given that obtaining healthy samples from the same tissue as the tumor is limited by factors such as availability or potential influence by the tumor activity and transcriptional profile of the surrounding tissue. As an alternative, Laumont *et al.* used RNA-seq data from thymic epithelial cells as a normal control, obtained from 3-month-old to 7-year-old patients who underwent corrective cardiovascular surgery (Laumont et al., 2018). However, these cells are hard to obtain and a comprehensive RNA-seq data base of such cells would need to be established and characterized first, before implementation into routine procedures. To mitigate the risk of false positive RNA variants originating from SNPs, a methodology that combines tumor RNA-seq with normal WES data was employed, proven to be effective in RNA variant calling (Hashimoto et al., 2021). This approach helped exclude common population SNPs, however, it did not control completely for false positive RNA variants. As the RNA variant data set is subsequently matched to another source of patient data for neoantigen candidate identification, namely the immunopeptidomics data, this step was seen as a biological cross validation further excluding false positive variants. Of note, due to this subsequent cross-validation of the variant data, less stringent mutation calling algorithms for both RNA and DNA variant detection were used. While this approach aimed to expand the search space for potential neoantigen identification, it also introduced the possibility that false positive hits might not have been completely excluded.

An additional validation step of the RNA variants would need to be implemented in future studies that could include searching the GTEx data base, to search for the occurrence and tissue distribution of all identified RNA editing events/variants. Such analysis could give a better understanding about the tumor specificity or tumor association of RNA editing events, especially for those shared between several patients, and could therefore be used to pre-filter the RNA variant data even before neoantigen identification. Additionally, RNA sequencing of healthy tissues such as PBMCs of each patient could be integrated to account for rare, patient-specific RNA editing events.

4.3.2 Proteogenomic analysis – a double edges sword

The proteogenomics-based neoantigen identification pipeline was not only improved by adding RNA-seq data on the genomic level but also by improvements on the peptidomic level. By using pFIND instead of MaxQuant and by also integrating Prosit, a rescoring algorithm that improves peptide-spectra matching, we were able to optimize the pipeline in terms of speed but most importantly in terms of numbers of identified antigens. This was seen for the previously published patient Mel15 (Bassani-Sternberg et al., 2016), where the improvements within this thesis identified 38-times more neoantigen candidates.

In the ImmuNEO MASTER cohort, in total 90 neoantigen candidates were successfully identified in samples of patients with several different entities (75% of samples) beyond melanoma. Comparisons to other publications in terms of frequency and number of neoantigens identified within a cross-entity cohort is not feasible, as few studies look at pan cancer cohorts or/and don’t use proteogenomics approaches. Interestingly, the number of neoantigens per patient did not depend on the entity and neoantigen candidates were identified also for patients with a low number of variants. By applying more loose filtering criteria with an FDR of 5 % on spectral level, more neoantigen candidates could be identified and were later also confirmed by the post-identification peptide verification. However, this peptide verification using SA and RT comparisons (Chong et al., 2020; Gessulat et al., 2019; Toprak et al., 2014; Verbruggen et al., 2021; D. Wang et al., 2019) also led to the exclusion of several neoantigen candidates (irrespective of their q values), as here potentially not the correct peptide sequence was identified. This data indicates that the strength of our neoantigen identification pipeline, the matching of MS-spectra to variants and therefore the direct identification of neoantigens from the tumor surface, is also its bottleneck.

4.3.2.1 Limitations and perspectives

Several factors influence the identification of neoantigens via proteogenomic strategies: HLA-bound peptides are on the one hand very diverse in their sequences, but on the other hand very similar in terms of length and amino acid composition. Also, each peptide is often found in very low abundance. The unique characteristics of neoantigens, such as their shorter length and hydrophobicity, present challenges for their separation using standard LC–MS/MS methods. These methods, typically optimized for tryptic proteome digests, may require a large amount of input material to achieve high sensitivity when dealing with neoantigens. Furthermore, also the algorithms used for HLA-I eluted peptide spectra interpretation are mainly build for typical proteomic studies that predominantly use trypsin-derived peptide spectra. This makes analysis of proteasomal degraded peptides more challenging and prone to errors. (Klaeger et al., 2021)

Therefore, improving MS-based neoantigen detection is crucial, and addressing four key points can contribute to advancements in this field.

(1) Optimizing artificial intelligence tools used fo matching and rescoring of MS spectra (like Prosit) holds the potential to enhance their capabilities for neoantigen discovery. In this thesis it is shown that, by combining pFIND with Prosit, 14 additional neoantigens were identified that would have been missed when solely using pFIND. Combining several tools and algorithms therefore also increases the potential for neoantigen identification, an approach also followed by *Chong et al.*, who combined two standard MS search tools, MaxQuant (Cox & Mann, 2008) and Comet (Eng et al., 2013), into a new tool called NewAnce (Chong et al., 2020). Examples of further algorithms or tools that could be explored and combined with our method are MHCquant (Bichmann et al., 2019), PeptideProphet (Keller et al., 2002; Ma et al., 2012) and ARTEMIS (Finton et al., 2021).

(2) Optimizing protocols for sample processing and pHLA-I immunoprecipitation may lead to a higher yield of detectable peptides. Also improved tumor tissue quality achieved by reduced ischemia times and direct freezing of samples after surgery would probably enhance peptide yield.

(3) One promising improvement also lies within the MS data acquisition strategy, as for example the implementation of data-independent acquisition for HLA immunopeptidome analysis improved peptide identification substantially (Pak et al., 2021).

(4) Enhancing the sensitivity of MS instruments has the potential to exert the biggest impact in the future (Caron et al., 2015), where already recent and promising progress has been made in the area of fractionation (Klaeger et al., 2021).

4.3.2.2 Proteogenomics vs. prediction

Although our proteogenomic pipeline substantially improved the identification of neoantigen candidates, the number of neoantigen candidates remains modest compared to the numerous hits found with epitope prediction models (Tran et al., 2015; Verdegaal et al., 2016). However, as much as 25% of the validated neoantigen candidates in this study induced a T cell response *in vitro*, surpassing the expectations set by standard epitope prediction approaches (Cohen et al., 2015; McGranahan et al., 2016; Tran et al., 2015). This drastically reduces the necessity for extensive and large-scale immunogenicity testing, which may be impractical in a clinical setting, and thus makes a proteogenomic approach more attractive. Furthermore, neoantigen identification via predication pipelines cannot proof actual tumor surface presentation of these peptides and thus lacks a crucial validation step towards therapy development.

However, several peptides from shared genomic alterations were solely found by prediction that might still represent interesting targets for immunotherapy and should be analyzed in future experiments. Also, an enrichment for predicted binders was seen when applying the peptide-verification criteria to all neoantigen candidates, indicating that binding prediction can be a good tool for peptide validation, although still some immunogenic validated neoantigen candidates were missed by the prediction algorithms.

As also the proteogenomics approach shows significant limitations due to the MS measurement as described above, a combinatory approach might be most feasible to circumvent the limitations of both approaches respectively and thus ensure successful identification and also prioritization of neoantigens in a tumor-agnostic manner.

4.3.3 Neoantigens from RNA sources

The fact that neoantigen candidates were found from protein coding but also from non-coding regions and most of the aberrant peptides arose due to RNA variants, emphasizes the great impact made by including RNA-seq data into the pipeline. The characteristics of all neoantigen candidates and their inducing variants also reflects the distribution seen for the total immunopeptidome and the genomic/transcriptomic data indicating no bias towards a specific peptide length, biotype or variant type.

Most interestingly, 79 neoantigen candidates were found from RNA variants, of which 9 were not covered on DNA level and thus could still be somatic and 70 had canonical read coverage on DNA level and thus could represent RNA editing induced aberrant peptides. However, as discussed before, although RNA editing events can be tumor specific, many such events have been described in healthy tissues as normal regulatory mechanisms. As no perfect RNA-seq normal control is in place, the potential of false positive hits is still given. To further validate the variants and their induced neoantigen candidates, an additional validation step was included for the neoantigen candidate variants. When checking for the abundance of the RNA variants within healthy tissue, 18 % were found to be canonical/not tumor specific. However, 70 % of variants were either not found at all in healthy tissues or only present at very low frequencies indicating a potential tumor specificity of these variants. Together with the peptide verification, 32 neoantigen candidates were set as validated of which still 24 originated from RNA only variants.

Of these 24 validated neoantigen candidates from RNA variants, nearly all (n = 21) could be RNA editing induced. Only three of these peptides were found in non-coding regions and might therefore just have been missed by WES on DNA level, however the remaining 18 neoantigen candidates are more likely induced by editing events. This data supports the findings of Zhang *et al*. and Zhou *et al.* who showed that peptides can arise from RNA editing and are presented on the tumor surface via HLA molecules (Zhang et al., 2018; Zhou et al., 2020). Furthermore, 11 of these 24 events were found in splice sites and led to the identification of intronic peptides. It was previously shown that for example A-to-I editing (seen as A-to-G in sequencing data) can produce or delete splice sites, leading to alternative splicing of RNAs and thus potential intron inclusions (Christofi & Zaravinos, 2019; Hsiao et al., 2018; Merlotti et al., 2023; S. J. Tang et al., 2020). However, as explained before, also "alternative mRNA editing" patterns such as C-to-U (mediated by APOBEC1), U-to-C, and G-to-A were observed (Christofi & Zaravinos, 2019; B. R. Rosenberg et al., 2011). The remaining 7 of the 24 RNA only variants were found in exons of protein coding regions and could represent additional RNA editing events leading to missense peptide sequences, that would need further characterization. This holds true for all validated neoantigen candidates if they were to be used as targets for immunotherapy.

Therefore, of course also the experimental assessment of these neoantigen candidates is of great importance to understand their biological and ultimately clinical significance. All identified neoantigen candidates were selected for immunogenicity assessment, which will be discussed in the next paragraph.

4.4 Immunogenicity assessment – suitable as neoantigen validation?

For immunogenicity testing and thereby ultimate biological validation of all 90 identified neoantigen candidates, immunogenicity assessment using ELIspot read out was implemented using autologous PBMC and TILs, however only 78 peptides were tested due to limitations in primary human material. Strikingly, 21 immunogenic neoantigen candidates (before validation) and more importantly 8 validated immunogenic neoantigens were identified in 6 patients within this thesis independent of tumor entity. Furthermore, no bias was observed towards one of the MS spectra matching tools and reactive peptides were found across several variant and transcript types.

Of course, the threshold for immunogenicity, that was set by us, also influences the number of immunogenic peptides strongly, as this is a rather subjective. We defined immunogenicity by comparing the SFU of the mutated peptide condition with an irrelevant/control peptide condition and set the threshold to a ratio of ≥ 2 (meaning at least twice as many spots for the mutated than control condition) and a delta of ≥ 50 (meaning at least 50 spots in the mutated condition if the control condition was 0), which accounts for a potential background/unspecific activation. This threshold is in accordance with data from Sahin *et al.*, that also set a spot count twice as high as the control as a positive signal (Sahin et al., 2017). However, their second threshold is a minimum of 5 spots per 100,000 cells, using already sorted CD8$^+$ and CD4$^+$ T cells. Thus, our threshold here is much stricter with 50 spots per up to 20,000 cells of unsorted PBMCs. As we used EBV-transformed target cells, that might cause some unspecific stimulation and thus higher background (see more discussion on this below), this was seen as adequate for this experimental set up. However, more reactive neoantigen candidates could have been identified with lower thresholds.

We could further show that the previously discussed RNA variants not only lead to the presentation of aberrant peptides as part of the immunopeptidome but that these neoantigen candidates are indeed immunogenic (7 reactive validated peptides from potential editing events). Neoantigen candidates from all kinds of sources, including RNA editing, have been previously described to be potentially immunogenic (Chong et al., 2020; Laumont et al., 2018; Smart et al., 2018; Zhang et al., 2018; Zhou et al., 2020), however still few studies evaluate the immunogenic potential of their neoantigen candidates by *in vitro* assays and even fewer studies actually can show reactivities. Chong *et al.* for example only identified five reactive neoantigens out of 786 peptides tested, which is a much smaller fraction as seen in this study. Interestingly, four of these reactive peptides were classified as TAAs from non-canonical regions and only one potential tumor-specific non-canonical neoantigen was found (Chong et al., 2020).

Importantly, immunogenicity of a peptide cannot be taken as an independent validation criterion. Although an immunological reactivity towards a neoantigen candidates is a positive indicator for more interesting clinical candidates, it does not eliminate the need for additional peptide and expression validation. 13 of the 20 immunogenic neoantigen candidates did not pass the in-depth validation, either because the variant was also expressed in more than 5 % of normal tissues or the peptide sequence was not verified. In the latter case, immunogenicity of the patients towards these peptides can be explained by the general ability of the immune system with its diverse TCR repertoire to recognize and react towards random non-self peptides. In the first case, autoreactivity towards a self-epitope could have occurred which was previously observed in cancer patients due to a suppressed immune system, treatment (side-) effects or excessive apoptosis (immunogenic cell death), that might even be beneficial (Berner et al., 2022; Zitvogel et al., 2021). Furthermore, an increase in RNA editing in systemic lupus erythematosus has been identified as a potential source of autoantigens (Roth et al., 2018), a mechanism that might also play a role in cancer. Thus, pure immunogenicity does not define or validate a neoantigen, but it is rather an additional factor that can be used to prioritize validated neoantigen candidates for therapy selection. Importantly, vice versa, non-reactive candidates cannot be categorized as less important or not clinically relevant, as several other factors might influence the identification of reactivities *in vitro* and an immunological response might have been missed due to biological and experimental limitations, as further elaborated below.

4.4.1 Limitations and perspectives

Reason for the low frequency of identified immunogenic neoantigens (8 out of 32 validated neoantigens) and reactive T cell clones are probably major challenges that arise when performing *in vitro* stimulations.

First, fresh PBMC or TILs need to be available for each patient, which is often not the case as several groups use already existing sequencing and proteomic data sets or use frozen or FFPE preserved tissues for analysis. These are logistically much easier to obtain than fresh tumor tissue. Also, many groups don’t have direct access to the patients for (periodic) fresh blood draws. Therefore, a big advantage of this study was the comprehensive sampling pipeline established within this thesis that gave access to such fresh material.

Second, the recovery of alive T cell from PBMC and even more so from TIL samples is often limited. By improving the sampling (e.g. shorter ischemia times), shipment and storage of samples, the fitness of cells could be enhanced but also demands for high numbers of trained personnel, immediate shipment or direct processing in specific on-site laboratories, which is often not feasible or too expensive.

Third, the choice of target cells for peptide presentation is crucial for a potent immune response as target cells need to be HLA-matched to the patient. Here, autologous LCLs can be used as target cells, however these potentially generate background responses towards EBV epitopes, as also observed in this study. Alternatively, cells transduced with the right (peptide-binding) HLA alleles could be used. For optimal testing those cells would need to either have all patient HLA alleles on their surface or, as that is not always feasible, prediction algorithms would need to be used to determine the most likely HLA molecule for each peptide. As prediction algorithms are still not fully accurate and don't work well for infrequent HLA alleles, the choice could fall on the "wrong" HLA allele. Thus, immunogenic peptides could be missed in any case due to a too high background response and overgrowth of EBV-specific T cells or the choice of the wrong presenting HLA allele.

Fourth, immunogenicity testing using autologous T cells carries the inherent risk of an undetectable immune response to the presented peptide due to various potential reasons. One cause can be the dysfunction of present T cells (Caushi et al., 2021), that thus cannot respond and proliferate anymore upon peptide stimulation. In contrast, overstimulation with excessively high peptide concentrations may on the other hand lead to the exhaustion and death of neoantigen-specific T cells. Additionally, the overgrowth of non-specific T cells due to very low frequencies of neoantigen-specific T cells in blood or TILs can lead to high background signals, making neoantigen-specific responses non-visible. he overgrowth of non-specific T cells may even deprive neoantigen-specific T cells of nutrients, leading to their death. Indeed, a previous report indicated that the *ex vivo* expansion of TILs may result in the depletion of T cell clones recognizing tumor neoantigens (Bobisse et al., 2018). Here especially the avidity/activation dynamics of the neoantigen-specific TCRs might play a role as discussed in Bräunlein et al., 2021.

Altogether, this suggests that some neoantigen candidates, which did not induce a detectable T cell response within this study, might in fact be immunogenic but were missed due to their biology or the experimental set-up. Also, these aspects, especially the last two, could have been the reason why no neoantigen-reactive T cell clone could be isolated from the bulk samples. These cells went through two steps of peptide stimulation which might have exhausted the reactive clones, especially when only present at low frequencies, leading to cell death rather than expansion.

Therefore, the development of more sensitive assays for the immunological validation of neoantigens is crucial and the integration of single-cell RNA and TCR sequencing appears promising to meet this demand (Caushi et al., 2021; Lowery et al., 2022; Lu et al., 2017).

4.4.2 Correlation with microenvironment

Within this thesis, the aim was to also identify potential biomarkers by combining the findings of all generated data sets. Several correlations of phenotypic, genomic and immunopeptidomic features with the patients' survival and the response to ICI have already been mentioned and discussed above. Additionally, it was observed within this thesis that patients with tumors showing an increased infiltration of T cells, with generally more CD3$^+$ T cells but also specifically more CD8$^+$ Teff, show a bigger likelihood to identify an immunogenic neoantigen candidate. These tumors might be more immunologically active and therefore have a higher frequency of tumor-reactive T cells within their TILs and blood. A recent study in Nature Cancer also shows such association of the immunopeptidomics landscape with T cell infiltration and immune recognition in lung cancer (Kraemer et al., 2023). Also, many TILs within these patients' tumors express inhibitory markers, suggesting that these cells might have been previously stimulated by the tumor (Brunet et al., 1988; Ishida et al., 1992; Waterhouse et al., 1995) and could recognize tumor-antigens also *in vitro*, in contrast to T cells not showing any marker expression that would be more immunologically silent. These patients might also benefit from ICI, as blocking the inhibitory receptors might increase the reactivities of these cells towards to tumor. Also, using the T cell infiltration and activation status as a biomarker for stratifying patients into those more likely to yield an immunogenic neoantigen and reactive T cell clone might be an option, however larger cohorts and more specific studies would be needed to further validate this finding.

4.5 Summary and future perspectives

In summary, proteogenomics identification of neoantigen candidates in a pan cancer cohort is feasible and leads to the identification of promising neoantigen candidates that could be analyzed/characterized in more detail for a personalized therapy approach.

By including RNA sequencing data for variant calling, a broader variety of neoantigen candidates was detected and thus the likelihood for neoantigen identification was improved, also for tumors with low mutational burden on DNA level. It is therefore recommended to include RNA sequencing data for proteogenomics but also prediction based neoantigen identification pipelines. However, further research on the biology of such peptides would need to be performed to understand the clinical impact of such antigens.

Based on the findings within this thesis the following pipeline would be proposed for improved proteogenomics neoantigen identification at this time: Use DNA and RNA sequencing data for variants calling and perform preliminary neoantigen identification applying relaxed criteria such as 5% FDR. Combining several bioinformatic peptide spectra matching tools for this step would also be beneficial to increase the potential for neoantigen identification and for peptide verification.

However, a post peptide verification as described within this thesis is recommended to exclude false positives before immunogenicity test. Also, when including RNA seq data, a tumor-specificity assessment of the variants would be needed prior to immunogenicity testing using normal tissue expression data analysis. For this, the use large repositories containing normal tissue expression data such as GTEx (Lonsdale et al., 2013) is a very powerful tool. Additionally, integrating total RNA-seq analysis of liquid biopsies from patients is recommended to exclude patient-specific variants and cover non-canonical regions often overlooked in databases, as much of the RNA-seq data from GTEx is mRNA-based. Checking the coverage of each variant within such data repositories is of great importance before using it for variant validation. Neoantigen candidates passing both validation criteria would be most promising for therapy development and could be focused on for immunogenicity assessment. Also, additionally integrating a predication-based pipeline might be a powerful tool to further expand the likelihood of neoantigen identification and for prioritization of peptides. This pipeline would enrich for neoantigen candidates with lower risks for on-target, off tumor toxicity, however not fully excluding the possibility for false positive detection.

In addition to the variant verification via normal tissue sequencing data, one could also think about integrating the same validation on peptide level. Repositories such as the HLA ligand atlas (Marcu et al., 2021), that include peptidomes of several benign human tissues, could be use also for comparison with the tumor immunopeptidome to exclude false positives. Furthermore, specific RNA editing calling tools could improve identification of such variants (John et al., 2017; Picardi & Pesole, 2013; Z. Wang et al., 2016), and by that exclusion of false positives, already during the process of variant calling. Also, RNA editing data bases such as REDIportal (Mansi et al., 2021) and RADAR (Ramaswami & Li, 2014) provide a summary of canonical editing sites, that could also be used as post-identification filter for RNA variants.

Finally, in order to improve the power of neoantigen identification and prioritization and to make it more applicable for clinical implementation, it is mandatory to harmonize identification strategies and pipelines across studies around the globe. The Tumor Neoantigen Selection Alliance (TESLA)(Wells et al., 2020) is one of such initiatives, which could help pave the way towards routine use in the clinics. However, ongoing research is needed in all fields, especially when defining candidates for clinical use.

5. References

Akbani, R., Akdemir, K. C., Aksoy, B. A., Albert, M., Ally, A., Amin, S. B., Arachchi, H., Arora, A., Auman, J. T., Ayala, B., Baboud, J., Balasundaram, M., Balu, S., Barnabas, N., Bartlett, J., Bartlett, P., Bastian, B. C., Baylin, S. B., Behera, M., … Zou, L. (2015). Genomic Classification of Cutaneous Melanoma. *Cell*, *161*(7), 1681–1696. https://doi.org/10.1016/J.CELL.2015.05.044

Akhoundova, D., & Rubin, M. A. (2022). Clinical application of advanced multi-omics tumor profiling: Shaping precision oncology of the future. *Cancer Cell*, *40*(9), 920–938. https://doi.org/10.1016/j.ccell.2022.08.011

Alexandrov, L. B., Nik-Zainal, S., Wedge, D. C., Aparicio, S. A. J. R., Behjati, S., Biankin, A. V., Bignell, G. R., Bolli, N., Borg, A., Børresen-Dale, A. L., Boyault, S., Burkhardt, B., Butler, A. P., Caldas, C., Davies, H. R., Desmedt, C., Eils, R., Eyfjörd, J. E., Foekens, J. A., … Stratton, M. R. (2013). Signatures of mutational processes in human cancer. *Nature*, *500*(7463), 415–421. https://doi.org/10.1038/nature12477

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Andreatta, M., & Nielsen, M. (2016). Gapped sequence alignment using artificial neural networks: Application to the MHC class i system. *Bioinformatics*, *32*(4), 511–517. https://doi.org/10.1093/bioinformatics/btv639

Bassani-Sternberg, M., Bräunlein, E., Klar, R., Engleitner, T., Sinitcyn, P., Audehm, S., Straub, M., Weber, J., Slotta-Huspenina, J., Specht, K., Martignoni, M. E., Werner, A., Hein, R., Busch, D. H., Peschel, C., Rad, R., Cox, J., Mann, M., & Krackhardt, A. M. (2016). Direct identification of clinically relevant neoepitopes presented on native human melanoma tissue by mass spectrometry. *Nature Communications*, *7*(May). https://doi.org/10.1038/ncomms13404

Bassani-Sternberg, M., Pletscher-Frankild, S., Jensen, L. J., & Mann, M. (2015). Mass Spectrometry of Human Leukocyte Antigen Class I Peptidomes Reveals Strong Effects of Protein Abundance and Turnover on Antigen Presentation. *Molecular & Cellular Proteomics*, *14*(3), 658–673. https://doi.org/10.1074/mcp.M114.042812

Bazak, L., Haviv, A., Barak, M., Jacob-Hirsch, J., Deng, P., Zhang, R., Isaacs, F. J., Rechavi, G., Li, J. B., Eisenberg, E., & Levanon, E. Y. (2014). A-to-I RNA editing occurs at over a hundred million genomic sites, located in a majority of human genes. *Genome Research*, *24*(3), 365–376. https://doi.org/10.1101/gr.164749.113

Berner, F., Bomze, D., Lichtensteiger, C., Walter, V., Niederer, R., Ali, O. H., Wyss, N., Bauer, J., Freudenmann, L. K., Marcu, A., Wolfschmitt, E. M., Haen, S., Gross, T., Abdou, M. T., Diem, S., Knöpfli, S., Sinnberg, T., Hofmeister, K., Cheng, H. W., … Flatz, L. (2022). Autoreactive napsin A-specific T cells are enriched in lung tumors and inflammatory lung lesions during immune checkpoint blockade. *Science Immunology*, *7*(75). https://doi.org/10.1126/SCIIMMUNOL.ABN9644

Berraondo, P., Sanmamed, M. F., Ochoa, M. C., Etxeberria, I., Aznar, M. A., Pérez-Gracia, J. L., Rodríguez-Ruiz, M. E., Ponz-Sarvise, M., Castañón, E., & Melero, I. (2019). Cytokines in clinical cancer immunotherapy. *British Journal of Cancer*, *120*(1), 6–15. https://doi.org/10.1038/s41416-018-0328-y

Bichmann, L., Nelde, A., Ghosh, M., Heumos, L., Mohr, C., Peltzer, A., Kuchenbecker, L., Sachsenberg, T., Walz, J. S., Stevanović, S., Rammensee, H. G., & Kohlbacher, O. (2019). MHCquant: Automated and Reproducible Data Analysis for Immunopeptidomics. *Journal of Proteome Research*, *18*(11), 3876–3884. https://doi.org/10.1021/acs.jproteome.9b00313

Bigot, J., Lalanne, A. I., Lucibello, F., Gueguen, P., Houy, A., Dayot, S., Ganier, O., Gilet, J., Tosello, J., Nemati, F., Pierron, G., Waterfall, J. J., Barnhill, R., Gardrat, S., Piperno-Neumann, S., Popova, T., Masson, V., Loew, D., Mariani, P., … Lantz, O. (2021). Splicing Patterns in SF3B1-Mutated Uveal Melanoma Generate Shared Immunogenic Tumor-Specific Neoepitopes. *Cancer Discovery*, *11*(8), 1938–1951. https://doi.org/10.1158/2159-8290.CD-20-0555

Bobisse, S., Genolet, R., Roberti, A., Tanyi, J. L., Racle, J., Stevenson, B. J., Iseli, C., Michel, A., Le Bitoux, M. A., Guillaume, P., Schmidt, J., Bianchi, V., Dangaj, D., Fenwick, C., Derré, L., Xenarios, I., Michielin, O., Romero, P., Monos, D. S., … Harari, A. (2018). Sensitive and frequent identification of high avidity neo-epitope specific CD8 + T cells in immunotherapy-naive ovarian cancer. *Nature Communications*, *9*(1). https://doi.org/10.1038/s41467-018-03301-0

Boyiadzis, M. M., Kirkwood, J. M., Marshall, J. L., Pritchard, C. C., Azad, N. S., & Gulley, J. L. (2018). Significance and implications of FDA approval of pembrolizumab for biomarker-defined disease. In *Journal for ImmunoTherapy of Cancer* (Vol. 6, Issue 1, pp. 1–7). BioMed Central. https://doi.org/10.1186/s40425-018-0342-x

Bräunlein, E., Lupoli, G., Abualrous, E. T., Krätzig, N. de A., Gosmann, D., Füchsl, F., Wietbrock, L., Lange, S., Engleitner, T., Lan, H., Audehm, S., Effenberger, M., Boxberg, M., Steiger, K., Chang, Y., Yu, K., Atay, C., Bassermann, F., Weichert, W., … Krackhardt, A. M. (2021). Spatial and temporal plasticity of neoantigen-specific T-cell responses bases on characteristics associated to antigen and TCR. *BioRxiv*, 2021.02.02.428777. https://doi.org/10.1101/2021.02.02.428777

Brunet, J. F., Denizot, F., Luciani, M. F., Roux-Dosseto, M., Suzan, M., Mattei, M. G., & Golstein, P. (1988). A new member of the immunoglobulin superfamily-CTLA-4. *Nature*, *328*(6127), 267–270. https://doi.org/10.1038/328267a0

Bruni, D., Angell, H. K., & Galon, J. (2020). The immune contexture and Immunoscore in cancer prognosis and therapeutic efficacy. In *Nature Reviews Cancer* (Vol. 20, Issue 11, pp. 662–680). Nat Rev Cancer. https://doi.org/10.1038/s41568-020-0285-7

Calis, J. J. A., Maybeno, M., Greenbaum, J. A., Weiskopf, D., De Silva, A. D., Sette, A., Keşmir, C., & Peters, B. (2013). Properties of MHC Class I Presented Peptides That Enhance Immunogenicity. *PLOS Computational Biology*, *9*(10), e1003266. https://doi.org/10.1371/JOURNAL.PCBI.1003266

Cameron, F., Whiteside, G., & Perry, C. (2011). Ipilimumab: First global approval. *Drugs*, *71*(8), 1093–1104. https://doi.org/10.2165/11594010-000000000-00000

Cantini, L., Zakeri, P., Hernandez, C., Naldi, A., Thieffry, D., Remy, E., & Baudot, A. (2021). Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer. *Nature Communications*, *12*(1). https://doi.org/10.1038/s41467-020-20430-7

Capietto, A. H., Hoshyar, R., & Delamarre, L. (2022). Sources of Cancer Neoantigens beyond Single-Nucleotide Variants. In *International Journal of Molecular Sciences* (Vol. 23, Issue 17, p. 10131). Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/ijms231710131

Carbone, D. P., Reck, M., Paz-Ares, L., Creelan, B., Horn, L., Steins, M., Felip, E., van den Heuvel, M. M., Ciuleanu, T.-E., Badin, F., Ready, N., Hiltermann, T. J. N., Nair, S., Juergens, R., Peters, S., Minenza, E., Wrangle, J. M., Rodriguez-Abreu, D., Borghaei, H., … Socinski, M. A. (2017). First-Line Nivolumab in Stage IV or Recurrent Non–Small-Cell Lung Cancer. *New England Journal of Medicine*, *376*(25), 2415–2426. https://doi.org/10.1056/nejmoa1613493

Caron, E., Kowalewski, D. J., Koh, C. C., Sturm, T., Schuster, H., & Aebersold, R. (2015). Analysis of major histocompatibility complex (MHC) immunopeptidomes using mass spectrometry. In *Molecular and Cellular Proteomics* (Vol. 14, Issue 12, pp. 3105–3117). American Society for Biochemistry and Molecular Biology. https://doi.org/10.1074/mcp.O115.052431

Carpenito, C., Milone, M. C., Hassan, R., Simonet, J. C., Lakhal, M., Suhoski, M. M., Varela-Rohena, A., Haines, K. M., Heitjan, D. F., Albelda, S. M., Carroll, R. G., Riley, J. L., Pastan, I., & June, C. H. (2009). Control of large, established tumor xenografts with genetically retargeted human T cells containing CD28 and CD137 domains. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(9), 3360–3365. https://doi.org/10.1073/pnas.0813101106

Caushi, J. X., Zhang, J., Ji, Z., Vaghasia, A., Zhang, B., Hsiue, E. H. C., Mog, B. J., Hou, W., Justesen, S., Blosser, R., Tam, A., Anagnostou, V., Cottrell, T. R., Guo, H., Chan, H. Y., Singh, D., Thapa, S., Dykema, A. G., Burman, P., … Smith, K. N. (2021). Transcriptional programs of neoantigen-specific TIL in anti-PD-1-treated lung cancers. *Nature*, *596*(7870), 126–132. https://doi.org/10.1038/s41586-021-03752-4

Cebon, J. S., McArthur, G. A., Chen, W., Davis, I. D., Gore, M. E., Thompson, J. F., Millward, M., Findlay, M. P. N., Dunbar, R., Ottensmeier, C. H. H., Venhaus, R. R., Nathan, P. D., Dalgleish, A. G., Cerundolo, V., Maraskovsky, E., Hopkins, W., Marsden, J., Smithers, B. M., Hersey, P., &

Evans, T. R. J. (2014). Randomized, double-blind phase II trial of NY-ESO-1 ISCOMATRIX vaccine and ISCOMATRIX adjuvant alone in patients with resected stage IIc, III, or IV malignant melanoma. *Journal of Clinical Oncology*, *32*(15_suppl), 9050–9050. https://doi.org/10.1200/jco.2014.32.15_suppl.9050

Cerami, E., Gao, J., Dogrusoz, U., Gross, B. E., Sumer, S. O., Aksoy, B. A., Jacobsen, A., Byrne, C. J., Heuer, M. L., Larsson, E., Antipin, Y., Reva, B., Goldberg, A. P., Sander, C., & Schultz, N. (2012). The cBio Cancer Genomics Portal: An open platform for exploring multidimensional cancer genomics data. *Cancer Discovery*, *2*(5), 401–404. https://doi.org/10.1158/2159-8290.CD-12-0095

Chakraborty, S., Hosen, M. I., Ahmed, M., & Shekhar, H. U. (2018). Onco-Multi-OMICS Approach: A New Frontier in Cancer Research. *BioMed Research International*, *2018*. https://doi.org/10.1155/2018/9836256

Chandran, S. S., Paria, B. C., Srivastava, A. K., Rothermel, L. D., Stephens, D. J., Dudley, M. E., Somerville, R., Wunderlich, J. R., Sherry, R. M., Yang, J. C., Rosenberg, S. A., & Kammula, U. S. (2015). Persistence of CTL clones targeting melanocyte differentiation antigens was insufficient to mediate significant melanoma regression in humans. *Clinical Cancer Research*, *21*(3), 534–543. https://doi.org/10.1158/1078-0432.CCR-14-2208

Chaplin, D. D. (2010). Overview of the Immune Response. *The Journal of Allergy and Clinical Immunology*, *125*(2 Suppl 2), S3. https://doi.org/10.1016/J.JACI.2009.12.980

Chen, C., Wei, M., Wang, C., Sun, D., Liu, P., Zhong, X., & Yu, W. (2020). Long noncoding RNA KCNQ1OT1 promotes colorectal carcinogenesis by enhancing aerobic glycolysis via hexokinase-2. *Aging*, *12*(12), 11685–11697. https://doi.org/10.18632/aging.103334

Cheng, R., Xu, Z., Luo, M., Wang, P., Cao, H., Jin, X., Zhou, W., Xiao, L., & Jiang, Q. (2022). Identification of alternative splicing-derived cancer neoantigens for mRNA vaccine development. *Briefings in Bioinformatics*, *23*(2). https://doi.org/10.1093/bib/bbab553

Chi, H., Liu, C., Yang, H., Zeng, W. F., Wu, L., Zhou, W. J., Niu, X. N., Ding, Y. H., Zhang, Y., Wang, R. M., Wang, Z. W., Chen, Z. L., Sun, R. X., Liu, T., Tan, G. M., Dong, M. Q., Xu, P., Zhang, P. H., & He, S. M. (2018). Open-pFind enables precise, comprehensive and rapid peptide identification in shotgun proteomics. In *bioRxiv* (p. 285395). bioRxiv. https://doi.org/10.1101/285395

Chong, C., Müller, M., Pak, H. S., Harnett, D., Huber, F., Grun, D., Leleu, M., Auger, A., Arnaud, M., Stevenson, B. J., Michaux, J., Bilic, I., Hirsekorn, A., Calviello, L., Simó-Riudalbas, L., Planet, E., Lubiński, J., Bryśkiewicz, M., Wiznerowicz, M., … Bassani-Sternberg, M. (2020). Integrated proteogenomic deep sequencing and analytics accurately identify non-canonical peptides in tumor immunopeptidomes. *Nature Communications*, *11*(1), 1–21. https://doi.org/10.1038/s41467-020-14968-9

Christofi, T., & Zaravinos, A. (2019). RNA editing in the forefront of epitranscriptomics and human health. *Journal of Translational Medicine*, *17*(1), 1–15. https://doi.org/10.1186/s12967-019-2071-4

ClinicalTrials.gov. (2023). https://clinicaltrials.gov/ct2/home

Cohen, C. J., Gartner, J. J., Horovitz-Fried, M., Shamalov, K., Trebska-McGowan, K., Bliskovsky, V. V., Parkhurst, M. R., Ankri, C., Prickett, T. D., Crystal, J. S., Li, Y. F., El-Gamil, M., Rosenberg, S. A., & Robbins, P. F. (2015). Isolation of neoantigen-specific T cells from tumor and peripheral lymphocytes. *Journal of Clinical Investigation*, *125*(10), 3981–3991. https://doi.org/10.1172/JCI82416

Cole, D. J., Weil, D. P., Shilyansky, J., Custer, M., Kawakami, Y., Rosenberg, S. A., & Nishimura, M. I. (1995). Characterization of the Functional Specificity of a Cloned T-Cell Receptor Heterodimer Recognizing the MART-1 Melanoma Antigen. *Cancer Research*, *55*(4), 748–752.

Conarty, J. P., & Wieland, A. (2023). The Tumor-Specific Immune Landscape in HPV+ Head and Neck Cancer. *Viruses*, *15*(6), 1296. https://doi.org/10.3390/V15061296

Coudray, A., Battenhouse, A. M., Bucher, P., & Iyer, V. R. (2018). Detection and benchmarking of somatic mutations in cancer genomes using RNA-seq data. *PeerJ*, *2018*(7), e5362. https://doi.org/10.7717/PEERJ.5362/SUPP-6

Coulie, P. G., Lehmann, F., Lethé, B., Herman, J., Lurquin, C., Andrawiss, M., & Boon, T. (1995). A mutated intron sequence codes for an antigenic peptide recognized by cytolytic T lymphocytes on a human melanoma. *Proceedings of the National Academy of Sciences*, *92*(17), 7976–7980. https://doi.org/10.1073/PNAS.92.17.7976

Couzin-Frankel, J. (2013). Breakthrough of the year 2013: Cancer immunotherapy. *Science*, *342*(6165), 1432–1433. https://doi.org/10.1126/science.342.6165.1432

Cox, J., & Mann, M. (2008). MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, *26*(12), 1367–1372. https://doi.org/10.1038/nbt.1511

CTpedia. (2021). http://www.cta.lncc.br/

Dagogo-Jack, I., & Shaw, A. T. (2018). Tumour heterogeneity and resistance to cancer therapies. *Nature Reviews Clinical Oncology*, *15*(2), 81–94. https://doi.org/10.1038/nrclinonc.2017.166

Davis, A. A., & Patel, V. G. (2019). The role of PD-L1 expression as a predictive biomarker: An analysis of all US food and drug administration (FDA) approvals of immune checkpoint inhibitors. *Journal for ImmunoTherapy of Cancer*, *7*(1). https://doi.org/10.1186/s40425-019-0768-9

Dembić, Z., Haas, W., Weiss, S., Mccubrey, J., Kiefer, H., Von Boehmer, H., & Steinmetz, M. (1986). Transfer of specificity by murine α and β T-cell receptor genes. *Nature*, *320*(6059), 232–238. https://doi.org/10.1038/320232a0

Desiere, F., Deutsch, E. W., King, N. L., Nesvizhskii, A. I., Mallick, P., Eng, J., Chen, S., Eddes, J., Loevenich, S. N., & Aebersold, R. (2006). The PeptideAtlas project. *Nucleic Acids Research*, *34*(Database issue), D655–D658. https://doi.org/10.1093/nar/gkj040

Dhandapani, H., Jayakumar, H., Seetharaman, A., Singh, S. S., Ganeshrajah, S., Jagadish, N., Suri, A., Thangarajan, R., & Ramanathan, P. (2021). Dendritic cells matured with recombinant human sperm associated antigen 9 (rhSPAG9) induce CD4+, CD8+ T cells and activate NK cells: a potential candidate molecule for immunotherapy in cervical cancer. *Cancer Cell International*, *21*(1), 1–14. https://doi.org/10.1186/s12935-021-01951-7

Djebali, S., Davis, C. A., Merkel, A., Dobin, A., Lassmann, T., Mortazavi, A., Tanzer, A., Lagarde, J., Lin, W., Schlesinger, F., Xue, C., Marinov, G. K., Khatun, J., Williams, B. A., Zaleski, C., Rozowsky, J., Röder, M., Kokocinski, F., Abdelhamid, R. F., … Gingeras, T. R. (2012). Landscape of transcription in human cells. *Nature*, *489*(7414), 101–108. https://doi.org/10.1038/nature11233

Dong, F. (2021). Pan-Cancer Molecular Biomarkers: A Paradigm Shift in Diagnostic Pathology. In *Surgical Pathology Clinics* (Vol. 14, Issue 3, pp. 507–516). Surg Pathol Clin. https://doi.org/10.1016/j.path.2021.05.012

DTU Health Tech. (2022). *MHCMotifDecon - 1.0 - Services - DTU Health Tech*. https://services.healthtech.dtu.dk/service.php?MHCMotifDecon-1.0

Duan, F., Duitama, J., Al Seesi, S., Ayres, C. M., Corcelli, S. A., Pawashe, A. P., Blanchard, T., McMahon, D., Sidney, J., Sette, A., Baker, B. M., Mandoiu, I. I., & Srivastava, P. K. (2014). Genomic and bioinformatic profiling of mutational neoepitopes reveals new rules to predict anticancer immunogenicity. *Journal of Experimental Medicine*, *211*(11), 2231–2248. https://doi.org/10.1084/JEM.20141308

Duchrow, T., Shtatland, T., Guettler, D., Pivovarov, M., Kramer, S., & Weissleder, R. (2009). Enhancing navigation in biomedical databases by community voting and database-driven text classification. *BMC Bioinformatics*, *10*, 317. https://doi.org/10.1186/1471-2105-10-317

Dudley, M. E., Gross, C. A., Somerville, R. P. T., Hong, Y., Schaub, N. P., Rosati, S. F., White, D. E., Nathan, D., Restifo, N. P., Steinberg, S. M., Wunderlich, J. R., Kammula, U. S., Sherry, R. M., Yang, J. C., Phan, G. Q., Hughes, M. S., Laurencot, C. M., & Rosenberg, S. A. (2013). Randomized selection design trial evaluating CD8+-enriched versus unselected tumor-infiltrating lymphocytes for adoptive cell therapy for patients with melanoma. *Journal of Clinical Oncology*, *31*(17), 2152–2159. https://doi.org/10.1200/JCO.2012.46.6441

EGA European Genome-Phenome Archive. (n.d.). Retrieved 8 May 2023, from https://ega-archive.org/

Eggermont, A. M. M., Chiarion-Sileni, V., Grob, J.-J., Dummer, R., Wolchok, J. D., Schmidt, H., Hamid, O., Robert, C., Ascierto, P. A., Richards, J. M., Lebbé, C., Ferraresi, V., Smylie, M., Weber, J. S., Maio, M., Bastholt, L., Mortier, L., Thomas, L., Tahir, S., … Testori, A. (2016). Prolonged Survival in Stage III Melanoma with Ipilimumab Adjuvant Therapy. *New England Journal of Medicine*, *375*(19), 1845–1855. https://doi.org/10.1056/nejmoa1611299

ENA European Nucleotide Archive. (n.d.). Retrieved 8 May 2023, from https://www.ebi.ac.uk/ena/browser/home

Eng, J. K., Jahan, T. A., & Hoopmann, M. R. (2013). Comet: An open-source MS/MS sequence database search tool. *Proteomics*, *13*(1), 22–24. https://doi.org/10.1002/pmic.201200439

Engell-Noerregaard, L., Hansen, T. H., Andersen, M. H., Thor Straten, P., & Svane, I. M. (2009). Review of clinical studies on dendritic cell-based vaccination of patients with malignant melanoma: assessment of correlation between clinical response and vaccine parameters. *Cancer Immunology, Immunotherapy*, *58*(1), 1–14. https://doi.org/10.1007/S00262-008-0568-4

Ferrara, J., Reddy, P., & Paczesny, S. (2010). Immunotherapy through T-cell receptor gene transfer induces severe graft-versus-host disease. *Immunotherapy*, *2*(6), 791–794. https://doi.org/10.2217/IMT.10.73

Finton, K. A. K., Brusniak, M. Y., Jones, L. A., Lin, C., Fioré-Gartland, A. J., Brock, C., Gafken, P. R., & Strong, R. K. (2021). ARTEMIS: A Novel Mass-Spec Platform for HLA-Restricted Self and Disease-Associated Peptide Discovery. *Frontiers in Immunology*, *12*, 1284. https://doi.org/10.3389/fimmu.2021.658372

Freeman, G. J., Long, A. J., Iwai, Y., Bourque, K., Chernova, T., Nishimura, H., Fitz, L. J., Malenkovich, N., Okazaki, T., Byrne, M. C., Horton, H. F., Fouser, L., Carter, L., Ling, V., Bowman, M. R., Carreno, B. M., Collins, M., Wood, C. R., & Honjo, T. (2000). Engagement of the PD-1 immunoinhibitory receptor by a novel B7 family member leads to negative regulation of lymphocyte activation. *Journal of Experimental Medicine*, *192*(7), 1027–1034. https://doi.org/10.1084/jem.192.7.1027

Galon, J., Costes, A., Sanchez-Cabo, F., Kirilovsky, A., Mlecnik, B., Lagorce-Pagès, C., Tosolini, M., Camus, M., Berger, A., Wind, P., Zinzindohoué, F., Bruneval, P., Cugnenc, P. H., Trajanoski, Z., Fridman, W. H., & Pagès, F. (2006). Type, density, and location of immune cells within human colorectal tumors predict clinical outcome. *Science*, *313*(5795), 1960–1964. https://doi.org/10.1126/science.1129139

Garcia-Garijo, A., Fajardo, C. A., & Gros, A. (2019). Determinants for neoantigen identification. *Frontiers in Immunology*, *10*(JUN), 1392. https://doi.org/10.3389/FIMMU.2019.01392/BIBTEX

Gessulat, S., Schmidt, T., Zolg, D. P., Samaras, P., Schnatbaum, K., Zerweck, J., Knaute, T., Rechenberger, J., Delanghe, B., Huhmer, A., Reimer, U., Ehrlich, H. C., Aiche, S., Kuster, B., & Wilhelm, M. (2019). Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods*, *16*(6), 509–518. https://doi.org/10.1038/s41592-019-0426-7

Gonzalez-Galarza, F. F., McCabe, A., Santos, E. J. M. Dos, Jones, J., Takeshita, L., Ortega-Rivera, N. D., Cid-Pavon, G. M. D., Ramsbottom, K., Ghattaoraya, G., Alfirevic, A., Middleton, D., & Jones, A. R. (2020a). Allele frequency net database (AFND) 2020 update: gold-standard data classification, open access genotype data and new query tools. *Nucleic Acids Research*, *48*(D1), D783–D788. https://doi.org/10.1093/NAR/GKZ1029

Gonzalez-Galarza, F. F., McCabe, A., Santos, E. J. M. Dos, Jones, J., Takeshita, L., Ortega-Rivera, N. D., Cid-Pavon, G. M. D., Ramsbottom, K., Ghattaoraya, G., Alfirevic, A., Middleton, D., & Jones, A. R. (2020b). Allele frequency net database (AFND) 2020 update: Gold-standard data classification, open access genotype data and new query tools. *Nucleic Acids Research*, *48*(D1), D783–D788. https://doi.org/10.1093/nar/gkz1029

Gros, A., Parkhurst, M. R., Tran, E., Pasetto, A., Robbins, P. F., Ilyas, S., Prickett, T. D., Gartner, J. J., Crystal, J. S., Roberts, I. M., Trebska-Mcgowan, K., Wunderlich, J. R., Yang, J. C., & Rosenberg, S. A. (2016). Prospective identification of neoantigen-specific lymphocytes in the peripheral blood of melanoma patients. *Nature Medicine 2016 22:4*, *22*(4), 433–438. https://doi.org/10.1038/nm.4051

Gupta, R., Mehta, A., & Wajapeyee, N. (2022). Transcriptional determinants of cancer immunotherapy response and resistance. In *Trends in Cancer* (Vol. 8, Issue 5, pp. 404–415). Cell Press. https://doi.org/10.1016/j.trecan.2022.01.008

Haen, S. P., Löffler, M. W., Rammensee, H. G., & Brossart, P. (2020). Towards new horizons: characterization, classification and implications of the tumour antigenic repertoire. *Nature Reviews Clinical Oncology*, *17*(10), 595–610. https://doi.org/10.1038/s41571-020-0387-x

Han, L., Diao, L., Yu, S., Xu, X., Li, J., Zhang, R., Yang, Y., Werner, H. M. J., Eterovic, A. K., Yuan, Y., Li, J., Nair, N., Minelli, R., Tsang, Y. H., Cheung, L. W. T., Jeong, K. J., Roszik, J., Ju, Z., Woodman, S. E., … Liang, H. (2015). The Genomic Landscape and Clinical Relevance of A-to-I RNA Editing in Human Cancers. *Cancer Cell*, *28*(4), 515–528. https://doi.org/10.1016/j.ccell.2015.08.013

Hanahan, D., & Weinberg, R. A. (2000). The hallmarks of cancer. *Cell*, *100*(1), 57–70. https://doi.org/10.1016/S0092-8674(00)81683-9

Hanahan, D., & Weinberg, R. A. (2011). Hallmarks of cancer: The next generation. *Cell*, *144*(5), 646–674. https://doi.org/10.1016/j.cell.2011.02.013

Hashimoto, S., Noguchi, E., Bando, H., Miyadera, H., Morii, W., Nakamura, T., & Hara, H. (2021). Neoantigen prediction in human breast cancer using RNA sequencing data. *Cancer Science*, *112*(1), 465–475. https://doi.org/10.1111/cas.14720

Havel, J. J., Chowell, D., & Chan, T. A. (2019). The evolving landscape of biomarkers for checkpoint inhibitor immunotherapy. In *Nature Reviews Cancer* (Vol. 19, Issue 3, pp. 133–150). NIH Public Access. https://doi.org/10.1038/s41568-019-0116-x

Heo, Y. J., Hwa, C., Lee, G. H., Park, J. M., & An, J. Y. (2021). Integrative multi-omics approaches in cancer research: From biological networks to clinical subtypes. *Molecules and Cells*, *44*(7), 433–443. https://doi.org/10.14348/molcells.2021.0042

Hiam-Galvez, K. J., Allen, B. M., & Spitzer, M. H. (2021). Systemic immunity in cancer. In *Nature Reviews Cancer* (Vol. 21, Issue 6, pp. 345–359). Nature Publishing Group. https://doi.org/10.1038/s41568-021-00347-z

Hilf, N., Kuttruff-Coqui, S., Frenzel, K., Bukur, V., Stevanović, S., Gouttefangeas, C., Platten, M., Tabatabai, G., Dutoit, V., van der Burg, S. H., thor Straten, P., Martínez-Ricarte, F., Ponsati, B., Okada, H., Lassen, U., Admon, A., Ottensmeier, C. H., Ulges, A., Kreiter, S., … Wick, W. (2019). Actively personalized vaccination trial for newly diagnosed glioblastoma. *Nature*, *565*(7738), 240–245. https://doi.org/10.1038/s41586-018-0810-y

Hoadley, K. A., Yau, C., Wolf, D. M., Cherniack, A. D., Tamborero, D., Ng, S., Leiserson, M. D. M., Niu, B., McLellan, M. D., Uzunangelov, V., Zhang, J., Kandoth, C., Akbani, R., Shen, H., Omberg, L., Chu, A., Margolin, A. A., Van't Veer, L. J., Lopez-Bigas, N., … Zou, L. (2014). Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, *158*(4), 929–944. https://doi.org/10.1016/j.cell.2014.06.049

Horak, P., Heining, C., Kreutzfeldt, S., Hutter, B., Mock, A., Hüllein, J., Fröhlich, M., Uhrig, S., Jahn, A., Rump, A., Gieldon, L., Möhrmann, L., Hanf, D., Teleanu, V., Heilig, C. E., Lipka, D. B., Allgäuer, M., Ruhnke, L., Laßmann, A., … Fröhling, S. (2021). Comprehensive genomic and transcriptomic analysis for guiding therapeutic decisions in patients with rare cancers. *Cancer Discovery*, *11*(11), 2780–2795. https://doi.org/10.1158/2159-8290.CD-21-0126

Hsiao, Y. H. E., Bahn, J. H., Yang, Y., Lin, X., Tran, S., Yang, E. W., Quinones-Valdez, G., & Xiao, X. (2018). RNA editing in nascent RNA affects pre-mRNA splicing. *Genome Research*, *28*(6), 812–823. https://doi.org/10.1101/gr.231209.117

Huang, J., El-Gamil, M., Dudley, M. E., Li, Y. F., Rosenberg, S. A., & Robbins, P. F. (2004). T Cells Associated with Tumor Regression Recognize Frameshifted Products of the CDKN2A Tumor Suppressor Gene Locus and a Mutated HLA Class I Gene Product. *The Journal of Immunology*, *172*(10), 6057–6064. https://doi.org/10.4049/JIMMUNOL.172.10.6057

Hudecek, M., Lupo-Stanghellini, M. T., Kosasih, P. L., Sommermeyer, D., Jensen, M. C., Rader, C., & Riddell, S. R. (2013). Receptor affinity and extracellular domain modifications affect tumor recognition by ROR1-specific chimeric antigen receptor T cells. *Clinical Cancer Research*, *19*(12), 3153–3164. https://doi.org/10.1158/1078-0432.CCR-13-0330

IEDB.org: Free epitope database and prediction resource. (2022). http://www.iedb.org/

Ishida, Y., Agata, Y., Shibahara, K., & Honjo, T. (1992). Induced expression of PD-1, a novel member of the immunoglobulin gene superfamily, upon programmed cell death. *EMBO Journal*, *11*(11), 3887–3895. https://doi.org/10.1002/j.1460-2075.1992.tb05481.x

Jiang, W., He, Y., He, W., Wu, G., Zhou, X., Sheng, Q., Zhong, W., Lu, Y., Ding, Y., Lu, Q., Ye, F., & Hua, H. (2021). Exhausted CD8+T Cells in the Tumor Immune Microenvironment: New Pathways to Therapy. In *Frontiers in Immunology* (Vol. 11, p. 3739). Frontiers Media S.A. https://doi.org/10.3389/fimmu.2020.622509

John, D., Weirick, T., Dimmeler, S., & Uchida, S. (2017). RNAEditor: easy detection of RNA editing events and the introduction of editing islands. *Briefings in Bioinformatics*, *18*(6), 993–1001. https://doi.org/10.1093/BIB/BBW087

Johnson, L. A., Morgan, R. A., Dudley, M. E., Cassard, L., Yang, J. C., Hughes, M. S., Kammula, U. S., Royal, R. E., Sherry, R. M., Wunderlich, J. R., Lee, C. C. R., Restifo, N. P., Schwarz, S. L., Cogdill, A. P., Bishop, R. J., Kim, H., Brewer, C. C., Rudy, S. F., VanWaes, C., … Rosenberg, S. A. (2009). Gene therapy with human and mouse T-cell receptors mediates cancer regression and targets normal tissues expressing cognate antigen. *Blood*, *114*(3), 535–546. https://doi.org/10.1182/BLOOD-2009-03-211714

Jørgensen, K. W., Rasmussen, M., Buus, S., & Nielsen, M. (2014). NetMHCstab – predicting stability of peptide–MHC-I complexes; impacts for cytotoxic T lymphocyte epitope discovery. *Immunology*, *141*(1), 18–26. https://doi.org/10.1111/IMM.12160

Julian, R., Savani, M., & Bauman, J. E. (2021). Immunotherapy Approaches in HPV-Associated Head and Neck Cancer. *Cancers*, *13*(23). https://doi.org/10.3390/CANCERS13235889

June, C. H., O'Connor, R. S., Kawalekar, O. U., Ghassemi, S., & Milone, M. C. (2018). CAR T cell immunotherapy for human cancer. *Science*, *359*(6382), 1361–1365. https://doi.org/10.1126/science.aar6711

Kaabinejadian, S., Barra, C., Alvarez, B., Yari, H., Hildebrand, W. H., & Nielsen, M. (2022). Accurate MHC Motif Deconvolution of Immunopeptidomics Data Reveals a Significant Contribution of DRB3, 4 and 5 to the Total DR Immunopeptidome. *Frontiers in Immunology*, *13*, 128. https://doi.org/10.3389/fimmu.2022.835454

Keller, A., Nesvizhskii, A. I., Kolker, E., & Aebersold, R. (2002). Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytical Chemistry*, *74*(20), 5383–5392. https://doi.org/10.1021/ac025747h

Kent, W. J. (2002). BLAT—The BLAST-Like Alignment Tool. *Genome Research*, *12*(4), 656–664. https://doi.org/10.1101/GR.229202

Keşmir, C., Nussbaum, A. K., Schild, H., Detours, V., & Brunak, S. (2002). Prediction of proteasome cleavage motifs by neural networks. *Protein Engineering*, *15*(4), 287–296. https://doi.org/10.1093/protein/15.4.287

Khurana, E., Fu, Y., Chakravarty, D., Demichelis, F., Rubin, M. A., & Gerstein, M. (2016). Role of non-coding sequence variants in cancer. *Nature Reviews. Genetics*, *17*(2), 93–108. https://doi.org/10.1038/NRG.2015.17

Klaeger, S., Apffel, A., Clauser, K. R., Sarkizova, S., Oliveira, G., Rachimi, S., Le, P. M., Tarren, A., Chea, V., Abelin, J. G., Braun, D. A., Ott, P. A., Keshishian, H., Hacohen, N., Keskin, D. B., Wu, C. J., & Carr, S. A. (2021). Optimized liquid and gas phase fractionation increases HLA-peptidome coverage for primary cell and tissue samples. *Molecular and Cellular Proteomics*, *20*, 100133. https://doi.org/10.1016/J.MCPRO.2021.100133

Klempner, S. J., Fabrizio, D., Bane, S., Reinhart, M., Peoples, T., Ali, S. M., Sokol, E. S., Frampton, G., Schrock, A. B., Anhorn, R., & Reddy, P. (2020). Tumor Mutational Burden as a Predictive Biomarker for Response to Immune Checkpoint Inhibitors: A Review of Current Evidence. *The Oncologist*, *25*(1). https://doi.org/10.1634/theoncologist.2019-0244

Kochenderfer, J. N., Dudley, M. E., Feldman, S. A., Wilson, W. H., Spaner, D. E., Maric, I., Stetler-Stevenson, M., Phan, G. Q., Hughes, M. S., Sherry, R. M., Yang, J. C., Kammula, U. S., Devillier, L., Carpenter, R., Nathan, D. A. N., Morgan, R. A., Laurencot, C., & Rosenberg, S. A. (2012). B-cell depletion and remissions of malignancy along with cytokine-associated toxicity in a clinical trial of anti-CD19 chimeric-antigen-receptor-transduced T cells. *Blood*, *119*(12), 2709–2720. https://doi.org/10.1182/blood-2011-10-384388

Kraemer, A. I., Chong, C., Huber, F., Pak, H. S., Stevenson, B. J., Müller, M., Michaux, J., Altimiras, E. R., Rusakiewicz, S., Simó-Riudalbas, L., Planet, E., Wiznerowicz, M., Dagher, J., Trono, D., Coukos, G., Tissot, S., & Bassani-Sternberg, M. (2023). The immunopeptidome landscape associated with T cell infiltration, inflammation and immune editing in lung cancer. *Nature Cancer*, *4*(5), 608–628. https://doi.org/10.1038/s43018-023-00548-5

Krug, K., Jaehnig, E. J., Satpathy, S., Blumenberg, L., Karpova, A., Anurag, M., Miles, G., Mertins, P., Geffen, Y., Tang, L. C., Heiman, D. I., Cao, S., Maruvka, Y. E., Lei, J. T., Huang, C., Kothadia, R. B., Colaprico, A., Birger, C., Wang, J., … Zimmerman, L. J. (2020). Proteogenomic Landscape of Breast Cancer Tumorigenesis and Targeted Therapy. *Cell*, *183*(5), 1436-1456.e31. https://doi.org/10.1016/j.cell.2020.10.036

Kruit, W. H. J., Suciu, S., Dreno, B., Mortier, L., Robert, C., Chiarion-Sileni, V., Maio, M., Testori, A., Dorval, T., Grob, J. J., Becker, J. C., Spatz, A., Eggermont, A. M. M., Louahed, J., Lehmann, F. F., Brichard, V. G., & Keilholz, U. (2013). Selection of immunostimulant AS15 for active immunization with MAGE-A3 protein: Results of a randomized phase II study of the European organisation for research and treatment of cancer melanoma group in metastatic melanoma. *Journal of Clinical Oncology*, *31*(19), 2413–2420. https://doi.org/10.1200/JCO.2012.43.7111

Lange, S., Engleitner, T., Mueller, S., Maresch, R., Zwiebel, M., González-Silva, L., Schneider, G., Banerjee, R., Yang, F., Vassiliou, G. S., Friedrich, M. J., Saur, D., Varela, I., & Rad, R. (2020). Analysis pipelines for cancer genome sequencing in mice. *Nature Protocols*, *15*(2), 266–315. https://doi.org/10.1038/s41596-019-0234-7

Larsen, M. V., Lundegaard, C., Lamberth, K., Buus, S., Lund, O., & Nielsen, M. (2007). Large-scale validation of methods for cytotoxic T-lymphocyte epitope prediction. *BMC Bioinformatics*, *8*. https://doi.org/10.1186/1471-2105-8-424

Laumont, C. M., Vincent, K., Hesnard, L., Audemard, É., Bonneil, É., Laverdure, J. P., Gendron, P., Courcelles, M., Hardy, M. P., Côté, C., Durette, C., St-Pierre, C., Benhammadi, M., Lanoix, J., Vobecky, S., Haddad, E., Lemieux, S., Thibault, P., & Perreault, C. (2018). Noncoding regions are the main source of targetable tumor-specific antigens. *Science Translational Medicine*, *10*(470). https://doi.org/10.1126/scitranslmed.aau5516

Le, D. T., Durham, J. N., Smith, K. N., Wang, H., Bartlett, B. R., Aulakh, L. K., Lu, S., Kemberling, H., Wilt, C., Luber, B. S., Wong, F., Azad, N. S., Rucki, A. A., Laheru, D., Donehower, R., Zaheer, A., Fisher, G. A., Crocenzi, T. S., Lee, J. J., … Diaz, L. A. (2017). Mismatch repair deficiency predicts response of solid tumors to PD-1 blockade. *Science*, *357*(6349), 409–413. https://doi.org/10.1126/science.aan6733

Le, D. T., Uram, J. N., Wang, H., Bartlett, B. R., Kemberling, H., Eyring, A. D., Skora, A. D., Luber, B. S., Azad, N. S., Laheru, D., Biedrzycki, B., Donehower, R. C., Zaheer, A., Fisher, G. A., Crocenzi, T. S., Lee, J. J., Duffy, S. M., Goldberg, R. M., de la Chapelle, A., … Diaz, L. A. (2015). PD-1 Blockade in Tumors with Mismatch-Repair Deficiency. *New England Journal of Medicine*, *372*(26), 2509–2520. https://doi.org/10.1056/nejmoa1500596

Leach, D. R., Krummel, M. F., & Allison, J. P. (1996). Enhancement of antitumor immunity by CTLA-4 blockade. *Science*, *271*(5256), 1734–1736. https://doi.org/10.1126/science.271.5256.1734

Leffers, N., Gooden, M. J. M., de Jong, R. A., Hoogeboom, B.-N., ten Hoor, K. A., Hollema, H., Boezen, H. M., van der Zee, A. G. J., Daemen, T., & Nijman, H. W. (2008). Prognostic significance of tumor-infiltrating T-lymphocytes in primary and metastatic lesions of advanced stage ovarian cancer. *Cancer Immunology, Immunotherapy 2008 58:3*, *58*(3), 449–459. https://doi.org/10.1007/S00262-008-0583-5

Lehmann, B. D., Colaprico, A., Silva, T. C., Chen, J., An, H., Ban, Y., Huang, H., Wang, L., James, J. L., Balko, J. M., Gonzalez-Ericsson, P. I., Sanders, M. E., Zhang, B., Pietenpol, J. A., & Chen, X. S. (2021). Multi-omics analysis identifies therapeutic vulnerabilities in triple-negative breast cancer subtypes. *Nature Communications*, *12*(1), 1–18. https://doi.org/10.1038/s41467-021-26502-6

Lemery, S., Keegan, P., & Pazdur, R. (2017). First FDA Approval Agnostic of Cancer Site - When a Biomarker Defines the Indication. *The New England Journal of Medicine*, *377*(15), 1409–1412. https://doi.org/10.1056/NEJMP1709968

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *Q-Bio.GN*.

Lindskrog, S. V., Prip, F., Lamy, P., Taber, A., Groeneveld, C. S., Birkenkamp-Demtröder, K., Jensen, J. B., Strandgaard, T., Nordentoft, I., Christensen, E., Sokac, M., Birkbak, N. J., Maretty, L., Hermann, G. G., Petersen, A. C., Weyerer, V., Grimm, M. O., Horstmann, M., Sjödahl, G., … Dyrskjøt, L. (2021). An integrated multi-omics analysis identifies prognostic molecular subtypes of non-muscle-invasive bladder cancer. *Nature Communications*, *12*(1), 1–18. https://doi.org/10.1038/s41467-021-22465-w

Litchfield, K., Reading, J. L., Puttick, C., Thakkar, K., Abbosh, C., Bentham, R., Watkins, T. B. K., Rosenthal, R., Biswas, D., Rowan, A., Lim, E., Al Bakir, M., Turati, V., Guerra-Assunção, J. A., Conde, L., Furness, A. J. S., Saini, S. K., Hadrup, S. R., Herrero, J., … Swanton, C. (2021). Meta-analysis of tumor- and T cell-intrinsic mechanisms of sensitization to checkpoint inhibition. *Cell*, *184*(3), 596-614.e14. https://doi.org/10.1016/j.cell.2021.01.002

Liu, Y., Sethi, N. S., Hinoue, T., Schneider, B. G., Cherniack, A. D., Sanchez-Vega, F., Seoane, J. A., Farshidfar, F., Bowlby, R., Islam, M., Kim, J., Chatila, W., Akbani, R., Kanchi, R. S., Rabkin, C. S., Willis, J. E., Wang, K. K., McCall, S. J., Mishra, L., … Laird, P. W. (2018). Comparative Molecular Analysis of Gastrointestinal Adenocarcinomas. *Cancer Cell*, *33*(4), 721-735.e8. https://doi.org/10.1016/J.CCELL.2018.03.010

London, M., & Gallo, E. (2020). Epidermal growth factor receptor (EGFR) involvement in epithelial-derived cancers and its current antibody-based immunotherapies. *Cell Biology International*, *44*(6), 1267–1282. https://doi.org/10.1002/CBIN.11340

Lonsdale, J., Thomas, J., Salvatore, M., Phillips, R., Lo, E., Shad, S., Hasz, R., Walters, G., Garcia, F., Young, N., Foster, B., Moser, M., Karasik, E., Gillard, B., Ramsey, K., Sullivan, S., Bridge, J., Magazine, H., Syron, J., … Moore, H. F. (2013). The Genotype-Tissue Expression (GTEx) project. *Nature Genetics 2013 45:6*, *45*(6), 580–585. https://doi.org/10.1038/ng.2653

Louis, C. U., Savoldo, B., Dotti, G., Pule, M., Yvon, E., Myers, G. D., Rossig, C., Russell, H. V., Diouf, O., Liu, E., Liu, H., Wu, M. F., Gee, A. P., Mei, Z., Rooney, C. M., Heslop, H. E., & Brenner, M. K. (2011). Antitumor activity and long-term fate of chimeric antigen receptor-positive T cells in patients with neuroblastoma. *Blood*, *118*(23), 6050–6056. https://doi.org/10.1182/blood-2011-05-354449

Lowery, F. J., Krishna, S., Yossef, R., Parikh, N. B., Chatani, P. D., Zacharakis, N., Parkhurst, M. R., Levin, N., Sindiri, S., Sachs, A., Hitscherich, K. J., Yu, Z., Vale, N. R., Lu, Y.-C., Zheng, Z., Jia, L., Gartner, J. J., Hill, V. K., Copeland, A. R., … Rosenberg, S. A. (2022). Molecular signatures of antitumor neoantigen-reactive T cells from metastatic human cancers. *Science*, eabl5447. https://doi.org/10.1126/science.abl5447

Lu, Y. C., Yao, X., Crystal, J. S., Li, Y. F., El-Gamil, M., Gross, C., Davis, L., Dudley, M. E., Yang, J. C., Samuels, Y., Rosenberg, S. A., & Robbins, P. F. (2014). Efficient Identification of Mutated Cancer Antigens Recognized by T Cells Associated with Durable Tumor Regressions. *Clinical Cancer Research*, *20*(13), 3401–3410. https://doi.org/10.1158/1078-0432.CCR-14-0433

Lu, Y. C., Zheng, Z., Robbins, P. F., Tran, E., Prickett, T. D., Gartner, J. J., Li, Y. F., Ray, S., Franco, Z., Bliskovsky, V., Fitzgerald, P. C., & Rosenberg, S. A. (2017). An Efficient Single-Cell RNA-Seq Approach to Identify Neoantigen-Specific T Cell Receptors. *Molecular Therapy*, *26*(2), 1–11. https://doi.org/10.1016/j.ymthe.2017.10.018

Lundegaard, C., Lund, O., Keşmir, C., Brunak, S., & Nielsen, M. (2007). Modeling the adaptive immune system: predictions and simulations. *Bioinformatics*, *23*(24), 3265–3275. https://doi.org/10.1093/BIOINFORMATICS/BTM471

Ma, K., Vitek, O., & Nesvizhskii, A. I. (2012). A statistical model-building perspective to identification of MS/MS spectra with PeptideProphet. *BMC Bioinformatics*, *13 Suppl 1*(16), 1–17. https://doi.org/10.1186/1471-2105-13-S16-S1

Mansi, L., Tangaro, M. A., Lo Giudice, C., Flati, T., Kopel, E., Schaffer, A. A., Castrignanò, T., Chillemi, G., Pesole, G., & Picardi, E. (2021). REDIportal: millions of novel A-to-I RNA editing events from thousands of RNAseq experiments. *Nucleic Acids Research*, *49*(D1), D1012–D1019. https://doi.org/10.1093/NAR/GKAA916

Marcu, A., Bichmann, L., Kuchenbecker, L., Kowalewski, D. J., Freudenmann, L. K., Backert, L., Mühlenbruch, L., Szolek, A., Lübke, M., Wagner, P., Engler, T., Matovina, S., Wang, J., Hauri-

Hohl, M., Martin, R., Kapolou, K., Walz, J. S., Velz, J., Moch, H., … Neidert, M. C. (2021). HLA Ligand Atlas: A benign reference of HLA-presented peptides to improve T-cell-based cancer immunotherapy. *Journal for ImmunoTherapy of Cancer*, *9*(4), e002071. https://doi.org/10.1136/jitc-2020-002071

Marcus, L., Fashoyin-Aje, L. A., Donoghue, M., Yuan, M., Rodriguez, L., Gallagher, P. S., Philip, R., Ghosh, S., Theoret, M. R., Beaver, J. A., Pazdur, R., & Lemery, S. J. (2021). FDA approval summary: Pembrolizumab for the treatment of tumor mutational burden-high solid tumors. *Clinical Cancer Research*, *27*(17), 4685–4689. https://doi.org/10.1158/1078-0432.CCR-21-0327

Martinuzzi, E., Afonso, G., Gagnerault, M. C., Naselli, G., Mittag, D., Combadière, B., Boitard, C., Chaput, N., Zitvogel, L., Harrison, L. C., & Mallone, R. (2011a). acDCs enhance human antigen-specific T-cell responses. *Blood*, *118*(8), 2128–2137. https://doi.org/10.1182/blood-2010-12-326231

Martinuzzi, E., Afonso, G., Gagnerault, M. C., Naselli, G., Mittag, D., Combadière, B., Boitard, C., Chaput, N., Zitvogel, L., Harrison, L. C., & Mallone, R. (2011b). acDCs enhance human antigen-specific T-cell responses. *Blood*, *118*(8), 2128–2137. https://doi.org/10.1182/blood-2010-12-326231

Massard, C., Gordon, M. S., Sharma, S., Rafii, S., Wainberg, Z. A., Luke, J., Curiel, T. J., Colon-Otero, G., Hamid, O., Sanborn, R. E., O'Donnell, P. H., Drakaki, A., Tan, W., Kurland, J. F., Rebelatto, M. C., Jin, X., Blake-Haskins, J. A., Gupta, A., & Segal, N. H. (2016). Safety and efficacy of durvalumab (MEDI4736), an anti-programmed cell death ligand-1 immune checkpoint inhibitor, in patients with advanced urothelial bladder cancer. *Journal of Clinical Oncology*, *34*(26), 3119–3125. https://doi.org/10.1200/JCO.2016.67.9761

Massard, C., Michiels, S., Ferté, C., Le Deley, M. C., Lacroix, L., Hollebecque, A., Verlingue, L., Ileana, E., Rosellini, S., Ammari, S., Ngo-Camus, M., Bahleda, R., Gazzah, A., Varga, A., Postel-Vinay, S., Loriot, Y., Even, C., Breuskin, I., Auger, N., … Soria, J. C. (2017). High-throughput genomics and clinical outcome in hard-to-treat advanced cancers: Results of the MOSCATO 01 trial. *Cancer Discovery*, *7*(6), 586–595. https://doi.org/10.1158/2159-8290.CD-16-1396

Maude, S. L., Laetsch, T. W., Buechner, J., Rives, S., Boyer, M., Bittencourt, H., Bader, P., Verneris, M. R., Stefanski, H. E., Myers, G. D., Qayed, M., De Moerloose, B., Hiramatsu, H., Schlis, K., Davis, K. L., Martin, P. L., Nemecek, E. R., Yanik, G. A., Peters, C., … Grupp, S. A. (2018). Tisagenlecleucel in Children and Young Adults with B-Cell Lymphoblastic Leukemia. *New England Journal of Medicine*, *378*(5), 439–448. https://doi.org/10.1056/nejmoa1709866

McGranahan, N., Furness, A. J. S., Rosenthal, R., Ramskov, S., Lyngaa, R., Saini, S. K., Jamal-Hanjani, M., Wilson, G. A., Birkbak, N. J., Hiley, C. T., Watkins, T. B. K., Shafi, S., Murugaesu, N., Mitter, R., Akarca, A. U., Linares, J., Marafioti, T., Henry, J. Y., Van Allen, E. M., … Swanton, C. (2016). Clonal neoantigens elicit T cell immunoreactivity and sensitivity to immune checkpoint blockade. *Science*, *351*(6280), 1463–1469. https://doi.org/10.1126/science.aaf1490

Meng, X., Sun, X., Liu, Z., & He, Y. (2021). A novel era of cancer/testis antigen in cancer immunotherapy. *International Immunopharmacology*, *98*, 107889. https://doi.org/10.1016/J.INTIMP.2021.107889

Menyhárt, O., & Győrffy, B. (2021). Multi-omics approaches in cancer research with applications in tumor subtyping, prognosis, and diagnosis. *Computational and Structural Biotechnology Journal*, *19*, 949–960. https://doi.org/10.1016/J.CSBJ.2021.01.009

Merlotti, A., Sadacca, B., Arribas, Y. A., Ngoma, M., Burbage, M., Goudot, C., Houy, A., Rocañín-Arjó, A., Lalanne, A., Seguin-Givelet, A., Lefevre, M., Heurtebise-Chrétien, S., Baudon, B., Oliveira, G., Loew, D., Carrascal, M., Wu, C. J., Lantz, O., Stern, M. H., … Amigorena, S. (2023). Noncanonical splicing junctions between exons and transposable elements represent a source of immunogenic recurrent neo-antigens in patients with lung cancer. *Science Immunology*, *8*(80), eabm6359. https://doi.org/10.1126/sciimmunol.abm6359

Mertins, P., Mani, D. R., Ruggles, K. V., Gillette, M. A., Clauser, K. R., Wang, P., Wang, X., Qiao, J. W., Cao, S., Petralia, F., Kawaler, E., Mundt, F., Krug, K., Tu, Z., Lei, J. T., Gatza, M. L., Wilkerson, M., Perou, C. M., Yellapantula, V., … Carr, S. A. (2016). Proteogenomics connects somatic mutations to signalling in breast cancer. *Nature*, *534*(7605), 55–62. https://doi.org/10.1038/nature18003

Migden, M. R., Rischin, D., Schmults, C. D., Guminski, A., Hauschild, A., Lewis, K. D., Chung, C. H., Hernandez-Aya, L., Lim, A. M., Chang, A. L. S., Rabinowits, G., Thai, A. A., Dunn, L. A., Hughes, B. G. M., Khushalani, N. I., Modi, B., Schadendorf, D., Gao, B., Seebach, F., … Fury, M. G. (2018). PD-1 Blockade with Cemiplimab in Advanced Cutaneous Squamous-Cell Carcinoma. *New England Journal of Medicine*. https://doi.org/10.1056/NEJMOA1805131/SUPPL_FILE/NEJMOA1805131_DISCLOSURES.PDF

Morgan, R. A., Chinnasamy, N., Abate-Daga, D., Gros, A., Robbins, P. F., Zheng, Z., Dudley, M. E., Feldman, S. A., Yang, J. C., Sherry, R. M., Phan, G. Q., Hughes, M. S., Kammula, U. S., Miller, A. D., Hessman, C. J., Stewart, A. A., Restifo, N. P., Quezado, M. M., Alimchandani, M., … Rosenberg, S. A. (2013). Cancer regression and neurological toxicity following anti-MAGE-A3 TCR gene therapy. *Journal of Immunotherapy (Hagerstown, Md. : 1997)*, *36*(2), 133–151. https://doi.org/10.1097/CJI.0B013E3182829903

Morgan, R. A., Dudley, M. E., Wunderlich, J. R., Hughes, M. S., Yang, J. C., Sherry, R. M., Royal, R. E., Topalían, S. L., Kammula, U. S., Restifo, N. P., Zheng, Z., Nahvi, A., De Vries, C. R., Rogers-Freezer, L. J., Mavroukakis, S. A., & Rosenberg, S. A. (2006). Cancer regression in patients after transfer of genetically engineered lymphocytes. *Science*, *314*(5796), 126–129. https://doi.org/10.1126/science.1129003

Nayak, A., Dutta, M., & Roychowdhury, A. (2021). Emerging oncogene ATAD2: Signaling cascades and therapeutic initiatives. *Life Sciences*, *276*, 119322. https://doi.org/10.1016/J.LFS.2021.119322

NCT02334735. (2022). *A Comparison of Matured Dendritic Cells and Montanide® in Study Subjects With High Risk of Melanoma Recurrence - NCT02334735*. ClinicalTrials.Gov. https://clinicaltrials.gov/ct2/show/NCT02334735

Neelapu, S. S., Locke, F. L., Bartlett, N. L., Lekakis, L. J., Miklos, D. B., Jacobson, C. A., Braunschweig, I., Oluwole, O. O., Siddiqi, T., Lin, Y., Timmerman, J. M., Stiff, P. J., Friedberg, J. W., Flinn, I. W., Goy, A., Hill, B. T., Smith, M. R., Deol, A., Farooq, U., … Go, W. Y. (2017). Axicabtagene Ciloleucel CAR T-Cell Therapy in Refractory Large B-Cell Lymphoma. *New England Journal of Medicine*, *377*(26), 2531–2544. https://doi.org/10.1056/nejmoa1707447

Nicora, G., Vitali, F., Dagliati, A., Geifman, N., & Bellazzi, R. (2020). Integrated Multi-Omics Analyses in Oncology: A Review of Machine Learning Methods and Tools. *Frontiers in Oncology*, *10*, 1030. https://doi.org/10.3389/fonc.2020.01030

Nielsen, M., Lund, O., Buus, S., & Lundegaard, C. (2010). MHC Class II epitope predictive algorithms. *Immunology*, *130*(3), 319–328. https://doi.org/10.1111/J.1365-2567.2010.03268.X

Nishikura, K. (2015). A-to-I editing of coding and non-coding RNAs by ADARs. *Nature Reviews Molecular Cell Biology 2015 17:2*, *17*(2), 83–96. https://doi.org/10.1038/nrm.2015.4

Obeng, E. A., Stewart, C., & Abdel-Wahab, O. (2019). Altered RNA processing in cancer pathogenesis and therapy. In *Cancer Discovery* (Vol. 9, Issue 11, pp. 1493–1510). American Association for Cancer Research. https://doi.org/10.1158/2159-8290.CD-19-0399

O'Donnell, T. J., Rubinsteyn, A., Bonsack, M., Riemer, A. B., Laserson, U., & Hammerbacher, J. (2018). MHCflurry: Open-Source Class I MHC Binding Affinity Prediction. *Cell Systems*, *7*(1), 129-132.e4. https://doi.org/10.1016/j.cels.2018.05.014

Okamoto, S., Mineno, J., Ikeda, H., Fujiwara, H., Yasukawa, M., Shiku, H., & Kato, I. (2009). Improved expression and reactivity of transduced tumor-specific TCRs in human lymphocytes by specific silencing of endogenous TCR. *Cancer Research*, *69*(23), 9003–9011. https://doi.org/10.1158/0008-5472.CAN-09-1450

Oliveira, G., Stromhaug, K., Klaeger, S., Kula, T., Frederick, D. T., Le, P. M., Forman, J., Huang, T., Li, S., Zhang, W., Xu, Q., Cieri, N., Clauser, K. R., Shukla, S. A., Neuberg, D., Justesen, S., MacBeath, G., Carr, S. A., Fritsch, E. F., … Wu, C. J. (2021). Phenotype, specificity and avidity of antitumour CD8+ T cells in melanoma. *Nature*, *596*(7870), 119–125. https://doi.org/10.1038/s41586-021-03704-y

Ordóñez-Reyes, C., Garcia-Robledo, J. E., Chamorro, D. F., Mosquera, A., Sussmann, L., Ruiz-Patiño, A., Arrieta, O., Zatarain-Barrón, L., Rojas, L., Russo, A., de Miguel-Perez, D., Rolfo, C., & Cardona, A. F. (2022). Bispecific Antibodies in Cancer Immunotherapy: A Novel Response to an Old Question. *Pharmaceutics*, *14*(6), 1243. https://doi.org/10.3390/PHARMACEUTICS14061243

Oshi, M., Asaoka, M., Tokumaru, Y., Yan, L., Matsuyama, R., Ishikawa, T., Endo, I., & Takabe, K. (2020). CD8 T Cell Score as a Prognostic Biomarker for Triple Negative Breast Cancer. *International Journal of Molecular Sciences 2020, Vol. 21, Page 6968*, *21*(18), 6968. https://doi.org/10.3390/IJMS21186968

Ott, P. A., Hu, Z., Keskin, D. B., Shukla, S. A., Sun, J., Bozym, D. J., Zhang, W., Luoma, A., Giobbie-Hurder, A., Peter, L., Chen, C., Olive, O., Carter, T. A., Li, S., Lieb, D. J., Eisenhaure, T., Gjini, E., Stevens, J., Lane, W. J., … Wu, C. J. (2017). An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature*, *547*(7662), 217–221. https://doi.org/10.1038/nature22991

Ouspenskaia, T., Law, T., Clauser, K. R., Klaeger, S., Sarkizova, S., Aguet, F., Li, B., Christian, E., Knisbacher, B. A., Le, P. M., Hartigan, C. R., Keshishian, H., Apffel, A., Oliveira, G., Zhang, W., Chen, S., Chow, Y. T., Ji, Z., Jungreis, I., … Regev, A. (2022). Unannotated proteins expand the MHC-I-restricted immunopeptidome in cancer. *Nature Biotechnology*, *40*(2), 209–217. https://doi.org/10.1038/s41587-021-01021-3

Pak, H. S., Michaux, J., Huber, F., Chong, C., Stevenson, B. J., Müller, M., Coukos, G., & Bassani-Sternberg, M. (2021). Sensitive immunopeptidomics by leveraging available large-scale multi-HLA spectral libraries, data-independent acquisition, and MS/MS prediction. *Molecular and Cellular Proteomics*, *20*. https://doi.org/10.1016/J.MCPRO.2021.100080

Palmer, D. H., Midgley, R. S., Mirza, N., Torr, E. E., Ahmed, F., Steele, J. C., Steven, N. M., Kerr, D. J., Young, L. S., & Adams, D. H. (2009). A phase II study of adoptive immunotherapy using dendritic cells pulsed with tumor lysate in patients with hepatocellular carcinoma. *Hepatology*, *49*(1), 124–132. https://doi.org/10.1002/hep.22626

Pan, J., Yu, H., Guo, Z., Liu, Q., Ding, M., Xu, K., & Mao, L. (2018). Emerging role of sperm-associated antigen 9 in tumorigenesis. *Biomedicine & Pharmacotherapy*, *103*, 1212–1216. https://doi.org/10.1016/J.BIOPHA.2018.04.168

Park, J., & Chung, Y. J. (2019). Identification of neoantigens derived from alternative splicing and RNA modification. *Genomics and Informatics*, *17*(3). https://doi.org/10.5808/GI.2019.17.3.e23

Park, J. H., Rivière, I., Gonen, M., Wang, X., Sénéchal, B., Curran, K. J., Sauter, C., Wang, Y., Santomasso, B., Mead, E., Roshal, M., Maslak, P., Davila, M., Brentjens, R. J., & Sadelain, M. (2018). Long-Term Follow-up of CD19 CAR Therapy in Acute Lymphoblastic Leukemia. *New England Journal of Medicine*, *378*(5), 449–459. https://doi.org/10.1056/nejmoa1709919

Parkhurst, M. R., Yang, J. C., Langan, R. C., Dudley, M. E., Nathan, D. A. N., Feldman, S. A., Davis, J. L., Morgan, R. A., Merino, M. J., Sherry, R. M., Hughes, M. S., Kammula, U. S., Phan, G. Q., Lim, R. M., Wank, S. A., Restifo, N. P., Robbins, P. F., Laurencot, C. M., & Rosenberg, S. A. (2011). T cells targeting carcinoembryonic antigen can mediate regression of metastatic colorectal cancer but induce severe transient colitis. *Molecular Therapy*, *19*(3), 620–626. https://doi.org/10.1038/mt.2010.272

Patel, S. P., & Kurzrock, R. (2015). PD-L1 expression as a predictive biomarker in cancer immunotherapy. In *Molecular Cancer Therapeutics* (Vol. 14, Issue 4, pp. 847–856). American Association for Cancer Research. https://doi.org/10.1158/1535-7163.MCT-14-0983

Pender, A., Titmuss, E., Pleasance, E. D., Fan, K. Y., Pearson, H., Brown, S. D., Grisdale, C. J., Topham, J. T., Shen, Y., Bonakdar, M., Taylor, G. A., Williamson, L. M., Mungall, K. L., Chuah, E., Mungall, A. J., Moore, R. A., Lavoie, J. M., Yip, S., Lim, H., … Laskin, J. (2021). Genome and transcriptome biomarkers of response to immune checkpoint inhibitors in advanced solid tumors. *Clinical Cancer Research*, *27*(1), 202–212. https://doi.org/10.1158/1078-0432.CCR-20-1163

Peng, X., Xu, X., Wang, Y., Hawke, D. H., Yu, S., Han, L., Zhou, Z., Mojumdar, K., Jeong, K. J., Labrie, M., Tsang, Y. H., Zhang, M., Lu, Y., Hwu, P., Scott, K. L., Liang, H., & Mills, G. B. (2018). A-to-I RNA Editing Contributes to Proteomic Diversity in Cancer. *Cancer Cell*, *33*(5), 817-828.e7. https://doi.org/10.1016/j.ccell.2018.03.026

PeptideAtlas. (2022). http://www.peptideatlas.org/

Perez-Riverol, Y., Bai, J., Bandla, C., García-Seisdedos, D., Hewapathirana, S., Kamatchinathan, S., Kundu, D. J., Prakash, A., Frericks-Zipper, A., Eisenacher, M., Walzer, M., Wang, S., Brazma, A., & Vizcaíno, J. A. (2022). The PRIDE database resources in 2022: A hub for mass spectrometry-based proteomics evidences. *Nucleic Acids Research*, *50*(D1), D543–D552. https://doi.org/10.1093/nar/gkab1038

Picardi, E., & Pesole, G. (2013). REDItools: high-throughput RNA editing detection made easy. *Bioinformatics (Oxford, England)*, *29*(14), 1813–1814. https://doi.org/10.1093/BIOINFORMATICS/BTT287

Pleasance, E., Bohm, A., Williamson, L. M., Nelson, J. M. T., Shen, Y., Bonakdar, M., Titmuss, E., Csizmok, V., Wee, K., Hosseinzadeh, S., Grisdale, C. J., Reisle, C., Taylor, G. A., Lewis, E., Jones, M. R., Bleile, D., Sadeghi, S., Zhang, W., Davies, A., … Laskin, J. (2022). Whole-genome and transcriptome analysis enhances precision cancer treatment options. *Annals of Oncology*, *33*(9), 939–949. https://doi.org/10.1016/j.annonc.2022.05.522

Poliseno, L., Marranci, A., & Pandolfi, P. P. (2015). Pseudogenes in human cancer. *Frontiers of Medicine*, *2*. https://doi.org/10.3389/fmed.2015.00068

Pollack, S. M. (2018). The potential of the CMB305 vaccine regimen to target NY-ESO-1 and improve outcomes for synovial sarcoma and myxoid/round cell liposarcoma patients. *Expert Review of Vaccines*, *17*(2), 107–114. https://doi.org/10.1080/14760584.2018.1419068

Postow, M. A., Sidlow, R., & Hellmann, M. D. (2018). Immune-Related Adverse Events Associated with Immune Checkpoint Blockade. *New England Journal of Medicine*, *378*(2), 158–168. https://doi.org/10.1056/nejmra1703481

Qin, S., Xu, L., Yi, M., Yu, S., Wu, K., & Luo, S. (2019). Novel immune checkpoint targets: moving beyond PD-1 and CTLA-4. *Molecular Cancer 2019 18:1*, *18*(1), 1–14. https://doi.org/10.1186/S12943-019-1091-2

Ramaswami, G., & Li, J. B. (2014). RADAR: a rigorously annotated database of A-to-I RNA editing. *Nucleic Acids Research*, *42*(Database issue), D109. https://doi.org/10.1093/NAR/GKT996

Rammensee, H.-G., Bachmann, J., Emmerich, N. N., Bachor, O. A., & Stevanovic, S. (1999). SYFPEITHI: database for MHC ligands and peptide motifs. *Immunogenetics*, *50*, 213–219.

Rathe, S. K., Popescu, F. E., Johnson, J. E., Watson, A. L., Marko, T. A., Moriarity, B. S., Ohlfest, J. R., & Largaespada, D. A. (2019). Identification of candidate neoantigens produced by fusion transcripts in human osteosarcomas. *Scientific Reports*, *9*(1), 1–11. https://doi.org/10.1038/s41598-018-36840-z

Raufaste-Cazavieille, V., Santiago, R., & Droit, A. (2022). Multi-omics analysis: Paving the path toward achieving precision medicine in cancer treatment and immuno-oncology. *Frontiers in Molecular Biosciences*, *9*. https://doi.org/10.3389/FMOLB.2022.962743

Reck, M., Rodríguez-Abreu, D., Robinson, A. G., Hui, R., Csőszi, T., Fülöp, A., Gottfried, M., Peled, N., Tafreshi, A., Cuffe, S., O'Brien, M., Rao, S., Hotta, K., Leiby, M. A., Lubiniecki, G. M., Shentu, Y., Rangwala, R., & Brahmer, J. R. (2016). Pembrolizumab versus Chemotherapy for PD-L1–Positive Non–Small-Cell Lung Cancer. *New England Journal of Medicine*, *375*(19), 1823–1833. https://doi.org/10.1056/nejmoa1606774

Restifo, N. P., Smyth, M. J., & Snyder, A. (2016). Acquired resistance to immunotherapy and future challenges. *Nature Reviews Cancer 2016 16:2*, *16*(2), 121–126. https://doi.org/10.1038/nrc.2016.2

Riaz, N., Havel, J. J., Makarov, V., Desrichard, A., Urba, W. J., Sims, J. S., Hodi, F. S., Martín-Algarra, S., Mandal, R., Sharfman, W. H., Bhatia, S., Hwu, W. J., Gajewski, T. F., Slingluff, C. L., Chowell, D., Kendall, S. M., Chang, H., Shah, R., Kuo, F., … Chan, T. A. (2017). Tumor and Microenvironment Evolution during Immunotherapy with Nivolumab. *Cell*, *171*(4), 934-949.e15. https://doi.org/10.1016/j.cell.2017.09.028

Rizvi, N. A., Hellmann, M. D., Snyder, A., Kvistborg, P., Makarov, V., Havel, J. J., Lee, W., Yuan, J., Wong, P., Ho, T. S., Miller, M. L., Rekhtman, N., Moreira, A. L., Ibrahim, F., Bruggeman, C., Gasmi, B., Zappasodi, R., Maeda, Y., Sander, C., … Chan, T. A. (2015). Mutational landscape determines sensitivity to PD-1 blockade in non-small cell lung cancer. *Science*, *348*(6230), 124–128. https://doi.org/10.1126/science.aaa1348

Robert, C., Ribas, A., Wolchok, J. D., Hodi, F. S., Hamid, O., Kefford, R., Weber, J. S., Joshua, A. M., Hwu, W. J., Gangadhar, T. C., Patnaik, A., Dronca, R., Zarour, H., Joseph, R. W., Boasberg, P., Chmielowski, B., Mateus, C., Postow, M. A., Gergich, K., … Daud, A. (2014). Anti-programmeddeath-receptor-1 treatment with pembrolizumab in ipilimumab-refractory advanced melanoma: A randomised dose-comparison cohort of a phase 1 trial. *The Lancet*, *384*(9948), 1109–1117. https://doi.org/10.1016/S0140-6736(14)60958-2

Rodon, J., Soria, J. C., Berger, R., Miller, W. H., Rubin, E., Kugel, A., Tsimberidou, A., Saintigny, P., Ackerstein, A., Braña, I., Loriot, Y., Afshar, M., Miller, V., Wunder, F., Bresson, C., Martini, J. F., Raynaud, J., Mendelsohn, J., Batist, G., … Kurzrock, R. (2019). Genomic and transcriptomic profiling expands precision cancer medicine: the WINTHER trial. *Nature Medicine*, *25*(5), 751–758. https://doi.org/10.1038/s41591-019-0424-4

Rooney, M. S., Shukla, S. A., Wu, C. J., Getz, G., & Hacohen, N. (2015). Molecular and genetic properties of tumors associated with local immune cytolytic activity. *Cell*, *160*(1–2), 48–61. https://doi.org/10.1016/j.cell.2014.12.033

Rosenberg, B. R., Hamilton, C. E., Mwangi, M. M., Dewell, S., & Papavasiliou, F. N. (2011). Transcriptome-wide sequencing reveals numerous APOBEC1 mRNA editing targets in transcript 3′ UTRs. *Nature Structural & Molecular Biology*, *18*(2), 230. https://doi.org/10.1038/NSMB.1975

Rosenberg, S. A., Yannelli, J. R., Yang, J. C., Topalian, S. L., Schwartzentruber, D. J., Weber, J. S., Parkinson, D. R., Seipp, C. A., Einhorn, J. H., & White, D. E. (1994). Treatment of patients with metastatic melanoma with autologous tumor-infiltrating lymphocytes and interleukin 2. *Journal of the National Cancer Institute*, *86*(15), 1159–1166. https://doi.org/10.1093/jnci/86.15.1159

Roth, S. H., Danan-Gotthold, M., Ben-Izhak, M., Rechavi, G., Cohen, C. J., Louzoun, Y., & Levanon, E. Y. (2018). Increased RNA Editing May Provide a Source for Autoantigens in Systemic Lupus Erythematosus. *Cell Reports*, *23*(1), 50–57. https://doi.org/10.1016/j.celrep.2018.03.036

Roth, S. H., Levanon, E. Y., & Eisenberg, E. (2019). Genome-wide quantification of ADAR adenosine-to-inosine RNA editing activity. *Nature Methods*, *16*(11), 1131–1138. https://doi.org/10.1038/s41592-019-0610-9

Ruiz Cuevas, M. V., Hardy, M. P., Hollý, J., Bonneil, É., Durette, C., Courcelles, M., Lanoix, J., Côté, C., Staudt, L. M., Lemieux, S., Thibault, P., Perreault, C., & Yewdell, J. W. (2021). Most non-canonical proteins uniquely populate the proteome or immunopeptidome. *Cell Reports*, *34*(10), 108815. https://doi.org/10.1016/j.celrep.2021.108815

Sadelain, M., Brentjens, R., & Rivière, I. (2013). The basic principles of chimeric antigen receptor (CAR) design. *Cancer Discovery*, *3*(4), 388. https://doi.org/10.1158/2159-8290.CD-12-0548

Saha, A., Kim, Y., Gewirtz, A. D. H., Jo, B., Gao, C., McDowell, I. C., Engelhardt, B. E., Battle, A., Aguet, F., Ardlie, K. G., Cummings, B. B., Gelfand, E. T., Getz, G., Hadley, K., Handsaker, R. E., Huang, K. H., Kashin, S., Karczewski, K. J., Lek, M., … Zhu, J. (2017). Co-expression networks reveal the tissue-specific regulation of transcription and splicing. *Genome Research*, *27*(11), 1843–1858. https://doi.org/10.1101/GR.216721.116

Sahin, U., Derhovanessian, E., Miller, M., Kloke, B. P., Simon, P., Löwer, M., Bukur, V., Tadmor, A. D., Luxemburger, U., Schrörs, B., Omokoko, T., Vormehr, M., Albrecht, C., Paruzynski, A., Kuhn, A. N., Buck, J., Heesch, S., Schreeb, K. H., Müller, F., … Türeci, Ö. (2017). Personalized RNA mutanome vaccines mobilize poly-specific therapeutic immunity against cancer. *Nature*, *547*(7662), 222–226. https://doi.org/10.1038/nature23003

Samstein, R. M., Lee, C. H., Shoushtari, A. N., Hellmann, M. D., Shen, R., Janjigian, Y. Y., Barron, D. A., Zehir, A., Jordan, E. J., Omuro, A., Kaley, T. J., Kendall, S. M., Motzer, R. J., Hakimi, A. A., Voss, M. H., Russo, P., Rosenberg, J., Iyer, G., Bochner, B. H., … Morris, L. G. T. (2019). Tumor mutational load predicts survival after immunotherapy across multiple cancer types. In *Nature Genetics* (Vol. 51, Issue 2, pp. 202–206). Nature Publishing Group. https://doi.org/10.1038/s41588-018-0312-8

Schumacher, T. N., & Schreiber, R. D. (2015). Neoantigens in cancer immunotherapy. In *Science* (Vol. 348, Issue 6230, pp. 69–74). American Association for the Advancement of Science. https://doi.org/10.1126/science.aaa4971

Schuster, S. J., Svoboda, J., Chong, E. A., Nasta, S. D., Mato, A. R., Anak, Ö., Brogdon, J. L., Pruteanu-Malinici, I., Bhoj, V., Landsburg, D., Wasik, M., Levine, B. L., Lacey, S. F., Melenhorst, J. J., Porter, D. L., & June, C. H. (2017). Chimeric Antigen Receptor T Cells in Refractory B-Cell Lymphomas.

New England Journal of Medicine, *377*(26), 2545–2554. https://doi.org/10.1056/nejmoa1708566

Schwitalle, Y., Kloor, M., Eiermann, S., Linnebacher, M., Kienle, P., Knaebel, H. P., Tariverdian, M., Benner, A., & von Knebel Doeberitz, M. (2008). Immune Response Against Frameshift-Induced Neopeptides in HNPCC Patients and Healthy HNPCC Mutation Carriers. *Gastroenterology*, *134*(4), 988–997. https://doi.org/10.1053/j.gastro.2008.01.015

Sebestyén, Z., Schooten, E., Sals, T., Zaldivar, I., San José, E., Alarcón, B., Bobisse, S., Rosato, A., Szöllősi, J., Gratama, J. W., Willemsen, R. A., & Debets, R. (2008). Human TCR that incorporate CD3zeta induce highly preferred pairing between TCRalpha and beta chains following gene transfer. *Journal of Immunology (Baltimore, Md. : 1950)*, *180*(11), 7736–7746. https://doi.org/10.4049/JIMMUNOL.180.11.7736

Shastry, M., Gupta, A., Chandarlapaty, S., Young, M., Powles, T., & Hamilton, E. (2023). Rise of Antibody-Drug Conjugates: The Present and Future. *American Society of Clinical Oncology Educational Book*, *43*(43), e390094. https://doi.org/10.1200/EDBK_390094

Smart, A. C., Margolis, C. A., Pimentel, H., He, M. X., Miao, D., Adeegbe, D., Fugmann, T., Wong, K. K., & Van Allen, E. M. (2018). Intron retention is a source of neoepitopes in cancer. *Nature Biotechnology*, *36*(11), 1056–1063. https://doi.org/10.1038/nbt.4239

Snyder, A., Makarov, V., Merghoub, T., Yuan, J., Zaretsky, J. M., Desrichard, A., Walsh, L. A., Postow, M. A., Wong, P., Ho, T. S., Hollmann, T. J., Bruggeman, C., Kannan, K., Li, Y., Elipenahli, C., Liu, C., Harbison, C. T., Wang, L., Ribas, A., … Chan, T. A. (2014). Genetic Basis for Clinical Response to CTLA-4 Blockade in Melanoma. *New England Journal of Medicine*, *371*(23), 2189–2199. https://doi.org/10.1056/nejmoa1406498

Somaiah, N., Block, M. S., Kim, J. W., Shapiro, G. I., Do, K. T., Hwu, P., Eder, J. P., Jones, R. L., Lu, H., Ter Meulen, J. H., Bohac, C., Chen, M., Hsu, F. J., Gnjatic, S., & Pollack, S. M. (2019). First-in-class, first-in-human study evaluating LV305, a dendritic-cell tropic lentiviral vector, in sarcoma and other solid tumors expressing NY-ESO-1. *Clinical Cancer Research*, *25*(19), 5808–5817. https://doi.org/10.1158/1078-0432.CCR-19-1025

Sterner, R. C., & Sterner, R. M. (2021). CAR-T cell therapy: current limitations and potential strategies. *Blood Cancer Journal*, *11*(4), 1–11. https://doi.org/10.1038/s41408-021-00459-7

Sterner, R. M., & Kenderian, S. S. (2020). Myeloid cell and cytokine interactions with chimeric antigen receptor-T-cell therapy: Implication for future therapies. *Current Opinion in Hematology*, *27*(1), 41–48. https://doi.org/10.1097/MOH.0000000000000559

Subramanian, I., Verma, S., Kumar, S., Jere, A., & Anamika, K. (2020). Multi-omics Data Integration, Interpretation, and Its Application. *Bioinformatics and Biology Insights*, *14*. https://doi.org/10.1177/1177932219899051

Szolek, A., Schubert, B., Mohr, C., Sturm, M., Feldhahn, M., & Kohlbacher, O. (2014). OptiType: Precision HLA typing from next-generation sequencing data. *Bioinformatics*, *30*(23), 3310–3316. https://doi.org/10.1093/bioinformatics/btu548

Tan, M. H., Li, Q., Shanmugam, R., Piskol, R., Kohler, J., Young, A. N., Liu, K. I., Zhang, R., Ramaswami, G., Ariyoshi, K., Gupte, A., Keegan, L. P., George, C. X., Ramu, A., Huang, N., Pollina, E. A., Leeman, D. S., Rustighi, A., Goh, Y. P. S., … Li, J. B. (2017). Dynamic landscape and regulation of RNA editing in mammals. *Nature 2017 550:7675*, *550*(7675), 249–254. https://doi.org/10.1038/nature24041

Tang, C., Li, L., Mo, T., Na, J., Qian, Z., Fan, D., Sun, X., Yao, M., Pan, L., Huang, Y., & Zhong, L. (2022). Oncolytic viral vectors in the era of diversified cancer therapy: from preclinical to clinical. *Clinical and Translational Oncology 2022 24:9*, *24*(9), 1682–1701. https://doi.org/10.1007/S12094-022-02830-X

Tang, S. J., Shen, H., An, O., Hong, H. Q., Li, J., Song, Y., Han, J., Tay, D. J. T., Ng, V. H. E., Bellido Molias, F., Leong, K. W., Pitcheshwar, P., Yang, H., & Chen, L. (2020). Cis- and trans-regulations of pre-mRNA splicing by RNA editing enzymes influence cancer development. *Nature Communications*, *11*(1), 1–17. https://doi.org/10.1038/s41467-020-14621-5

The Cancer Genome Atlas Research Network. (2013). Genomic and Epigenomic Landscapes of Adult De Novo Acute Myeloid Leukemia. *New England Journal of Medicine*, *368*(22), 2059–2074. https://doi.org/10.1056/nejmoa1301689

Thommen, D. S., & Schumacher, T. N. (2018). T Cell Dysfunction in Cancer. *Cancer Cell*, *33*(4), 547–562. https://doi.org/10.1016/j.ccell.2018.03.012

Tini, G., Marchetti, L., Priami, C., & Scott-Boyer, M. P. (2018). Multi-omics integration-A comparison of unsupervised clustering methodologies. *Briefings in Bioinformatics*, *20*(4), 1269–1279. https://doi.org/10.1093/bib/bbx167

Topalian, S. L., Drake, C. G., & Pardoll, D. M. (2015). Immune checkpoint blockade: A common denominator approach to cancer therapy. In *Cancer Cell* (Vol. 27, Issue 4, pp. 450–461). Cell Press. https://doi.org/10.1016/j.ccell.2015.03.001

Topalian, S. L., Hodi, F. S., Brahmer, J. R., Gettinger, S. N., Smith, D. C., McDermott, D. F., Powderly, J. D., Carvajal, R. D., Sosman, J. A., Atkins, M. B., Leming, P. D., Spigel, D. R., Antonia, S. J., Horn, L., Drake, C. G., Pardoll, D. M., Chen, L., Sharfman, W. H., Anders, R. A., … Sznol, M. (2012). Safety, Activity, and Immune Correlates of Anti–PD-1 Antibody in Cancer. *New England Journal of Medicine*, *366*(26), 2443–2454. https://doi.org/10.1056/nejmoa1200690

Topalian, S. L., Hodi, F. S., Brahmer, J. R., Gettinger, S. N., Smith, D. C., McDermott, D. F., Powderly, J. D., Sosman, J. A., Atkins, M. B., Leming, P. D., Spigel, D. R., Antonia, S. J., Drilon, A., Wolchok, J. D., Carvajal, R. D., McHenry, M. B., Hosein, F., Harbison, C. T., Grosso, J. F., & Sznol, M. (2019). Five-Year Survival and Correlates among Patients with Advanced Melanoma, Renal Cell Carcinoma, or Non-Small Cell Lung Cancer Treated with Nivolumab. *JAMA Oncology*, *5*(10), 1411–1420. https://doi.org/10.1001/jamaoncol.2019.2187

Toprak, U. H., Gillet, L. C., Maiolica, A., Navarro, P., Leitner, A., & Aebersold, R. (2014). Conserved peptide fragmentation as a benchmarking tool for mass spectrometers and a discriminating feature for targeted proteomics. *Molecular and Cellular Proteomics*, *13*(8), 2056–2071. https://doi.org/10.1074/mcp.O113.036475

Torres-Jiménez, J., Esteban-Villarrubia, J., & Ferreiro-Monteagudo, R. (2022). Precision Medicine in Metastatic Colorectal Cancer: Targeting ERBB2 (HER-2) Oncogene. *Cancers 2022, Vol. 14, Page 3718*, *14*(15), 3718. https://doi.org/10.3390/CANCERS14153718

Tran, E., Ahmadzadeh, M., Lu, Y. C., Gros, A., Turcotte, S., Robbins, P. F., Gartner, J. J., Zheng, Z., Li, Y. F., Ray, S., Wunderlich, J. R., Somerville, R. P., & Rosenberg, S. A. (2015). Immunogenicity of somatic mutations in human gastrointestinal cancers. *Science*, *350*(6266), 1387–1390. https://doi.org/10.1126/science.aad1253

Tran, E., Robbins, P. F., Lu, Y.-C., Prickett, T. D., Gartner, J. J., Jia, L., Pasetto, A., Zheng, Z., Ray, S., Groh, E. M., Kriley, I. R., & Rosenberg, S. A. (2016). T-Cell Transfer Therapy Targeting Mutant KRAS in Cancer. *New England Journal of Medicine*, *375*(23), 2255–2262. https://doi.org/10.1056/nejmoa1609279

Tretter, C., de Andrade Krätzig, N., Pecoraro, M., Lange, S., Seifert, P., von Frankenberg, C., Untch, J., Zuleger, G., Wilhelm, M., Zolg, D. P., Dreyer, F. S., Bräunlein, E., Engleitner, T., Uhrig, S., Boxberg, M., Steiger, K., Slotta-Huspenina, J., Ochsenreither, S., von Bubnoff, N., … Krackhardt, A. M. (2023). Proteogenomic analysis reveals RNA as a source for tumor-agnostic neoantigen identification. *Nature Communications 2023 14:1*, *14*(1), 1–22. https://doi.org/10.1038/s41467-023-39570-7

Uhlen, M., Zhang, C., Lee, S., Sjöstedt, E., Fagerberg, L., Bidkhori, G., Benfeitas, R., Arif, M., Liu, Z., Edfors, F., Sanli, K., Von Feilitzen, K., Oksvold, P., Lundberg, E., Hober, S., Nilsson, P., Mattsson, J., Schwenk, J. M., Brunnström, H., … Ponten, F. (2017). A pathology atlas of the human cancer transcriptome. *Science*, *357*(6352). https://doi.org/10.1126/science.aan2507

U.S. Food and Drug Administration. (2017). *Pembrolizumab prescribing information.* https://www.accessdata.fda.gov/drugsatfda_docs/label/2020/125514s084lbl.pdf.

U.S. Food and Drug Administration. (2018). *FDA approves larotrectinib for solid tumors with NTRK gene fusions*. Case Medical Research. https://doi.org/10.31525/fda1-ucm626720.htm

U.S. Food and Drug Administration. (2023). *U.S. Food and Drug Administration*. https://www.fda.gov/

Vaddepally, R. K., Kharel, P., Pandey, R., Garje, R., & Chandra, A. B. (2020). Review of indications of FDA-approved immune checkpoint inhibitors per NCCN guidelines with the level of evidence. *Cancers*, *12*(3), 738. https://doi.org/10.3390/cancers12030738

Van den Eynden, J., Jiménez-Sánchez, A., Miller, M. L., & Larsson, E. (2019). Lack of detectable neoantigen depletion signals in the untreated cancer genome. *Nature Genetics*, *51*(12), 1741–1748. https://doi.org/10.1038/s41588-019-0532-6

Verbruggen, S., Gessulat, S., Gabriels, R., Matsaroki, A., van de Voorde, H., Kuster, B., Degroeve, S., Martens, L., van Criekinge, W., Wilhelm, M., & Menschaert, G. (2021). Spectral prediction features as a solution for the search space size problem in proteogenomics. *Molecular and Cellular Proteomics*, *20*, 100076. https://doi.org/10.1016/J.MCPRO.2021.100076

Verdegaal, E. M. E., De Miranda, N. F. C. C., Visser, M., Harryvan, T., Van Buuren, M. M., Andersen, R. S., Hadrup, S. R., Van Der Minne, C. E., Schotte, R., Spits, H., Haanen, J. B. A. G., Kapiteijn, E. H. W., Schumacher, T. N., & Van Der Burg, S. H. (2016). Neoantigen landscape dynamics during human melanoma-T cell interactions. *Nature*, *536*(7614), 91–95. https://doi.org/10.1038/nature18945

Vita, R., Mahajan, S., Overton, J. A., Dhanda, S. K., Martini, S., Cantrell, J. R., Wheeler, D. K., Sette, A., & Peters, B. (2019). The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Research*, *47*(D1), D339–D343. https://doi.org/10.1093/nar/gky1006

Wang, D., Eraslan, B., Wieland, T., Hallström, B., Hopf, T., Zolg, D. P., Zecha, J., Asplund, A., Li, L., Meng, C., Frejno, M., Schmidt, T., Schnatbaum, K., Wilhelm, M., Ponten, F., Uhlen, M., Gagneur, J., Hahne, H., & Kuster, B. (2019). A deep proteome and transcriptome abundance atlas of 29 healthy human tissues. *Molecular Systems Biology*, *15*(2). https://doi.org/10.15252/msb.20188503

Wang, H., Chen, S., Wei, J., Song, G., & Zhao, Y. (2021). A-to-I RNA Editing in Cancer: From Evaluating the Editing Level to Exploring the Editing Effects. *Frontiers in Oncology*, *0*, 3372. https://doi.org/10.3389/FONC.2020.632187

Wang, Z., Lian, J., Li, Q., Zhang, P., Zhou, Y., Zhan, X., & Zhang, G. (2016). RES-Scanner: a software package for genome-wide identification of RNA-editing sites. *GigaScience*, *5*(1). https://doi.org/10.1186/S13742-016-0143-4

Waterhouse, P., Penninger, J. M., Timms, E., Wakeham, A., Shahinian, A., Lee, K. P., Thompson, C. B., Griesser, H., & Mak, T. W. (1995). Lymphoproliferative disorders with early lethality in mice deficient in CTLA-4. *Science*, *270*(5238), 985–988. https://doi.org/10.1126/science.270.5238.985

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Sander, C., Stuart, J. M., Chang, K., Creighton, C. J., Davis, C., Donehower, L., Drummond, J., Wheeler, D., Ally, A., Balasundaram, M., Birol, I., Butterfield, Y. S. N., Chu, A., … Kling, T. (2013). The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, *45*(10), 1113–1120. https://doi.org/10.1038/ng.2764

Wells, D. K., van Buuren, M. M., Dang, K. K., Hubbard-Lucey, V. M., Sheehan, K. C. F., Campbell, K. M., Lamb, A., Ward, J. P., Sidney, J., Blazquez, A. B., Rech, A. J., Zaretsky, J. M., Comin-Anduix, B., Ng, A. H. C., Chour, W., Yu, T. V., Rizvi, H., Chen, J. M., Manning, P., … Defranoux, N. A. (2020). Key Parameters of Tumor Epitope Immunogenicity Revealed Through a Consortium Approach Improve Neoantigen Prediction. *Cell*, *183*(3), 818-834.e13. https://doi.org/10.1016/j.cell.2020.09.015

Wilhelm, M., Zolg, D. P., Graber, M., Gessulat, S., Schmidt, T., Schnatbaum, K., Schwencke-Westphal, C., Seifert, P., de Andrade Krätzig, N., Zerweck, J., Knaute, T., Bräunlein, E., Samaras, P., Lautenbacher, L., Klaeger, S., Wenschuh, H., Rad, R., Delanghe, B., Huhmer, A., … Kuster, B. (2021). Deep learning boosts sensitivity of mass spectrometry-based immunopeptidomics. *Nature Communications*, *12*(1), 3346. https://doi.org/10.1038/s41467-021-23713-9

Xie, C., Yeo, Z. X., Wong, M., Piper, J., Long, T., Kirkness, E. F., Biggs, W. H., Bloom, K., Spellman, S., Vierra-Green, C., Brady, C., Scheuermann, R. H., Telenti, A., Howard, S., Brewerton, S., Turpaz, Y., & Venter, J. C. (2017). Fast and accurate HLA typing from short-read next-generation sequence data with xHLA. *Proceedings of the National Academy of Sciences of the United States of America*, *114*(30), 8059–8064. https://doi.org/10.1073/pnas.1707945114

Yang, H. D., & Nam, S. W. (2020). Pathogenic diversity of RNA variants and RNA variation-associated factors in cancer development. In *Experimental and Molecular Medicine* (Vol. 52, Issue 4, pp. 582–593). Nature Publishing Group. https://doi.org/10.1038/s12276-020-0429-6

Ye, Q., Song, D. G., Poussin, M., Yamamoto, T., Best, A., Li, C., Coukos, G., & Powell, D. J. (2014). CD137 accurately identifies and enriches for naturally occurring tumor-reactive T cells in tumor. *Clinical Cancer Research*, *20*(1), 44–55. https://doi.org/10.1158/1078-0432.CCR-13-0945

Zacharakis, N., Chinnasamy, H., Black, M., Xu, H., Lu, Y. C., Zheng, Z., Pasetto, A., Langhan, M., Shelton, T., Prickett, T., Gartner, J., Jia, L., Trebska-Mcgowan, K., Somerville, R. P., Robbins, P. F., Rosenberg, S. A., Goff, S. L., & Feldman, S. A. (2018). Immune recognition of somatic mutations leading to complete durable regression in metastatic breast cancer. *Nature Medicine*, *24*(6), 724–730. https://doi.org/10.1038/s41591-018-0040-8

Zaretsky, J. M., Garcia-Diaz, A., Shin, D. S., Escuin-Ordinas, H., Hugo, W., Hu-Lieskovan, S., Torrejon, D. Y., Abril-Rodriguez, G., Sandoval, S., Barthly, L., Saco, J., Homet Moreno, B., Mezzadra, R., Chmielowski, B., Ruchalski, K., Shintaku, I. P., Sanchez, P. J., Puig-Saus, C., Cherry, G., … Ribas, A. (2016). Mutations Associated with Acquired Resistance to PD-1 Blockade in Melanoma. *New England Journal of Medicine*, *375*(9), 819–829. https://doi.org/10.1056/nejmoa1604958

Zehir, A., Benayed, R., Shah, R. H., Syed, A., Middha, S., Kim, H. R., Srinivasan, P., Gao, J., Chakravarty, D., Devlin, S. M., Hellmann, M. D., Barron, D. A., Schram, A. M., Hameed, M., Dogan, S., Ross, D. S., Hechtman, J. F., DeLair, D. F., Yao, J. J., … Berger, M. F. (2017). Mutational landscape of metastatic cancer revealed from prospective clinical sequencing of 10,000 patients. *Nature Medicine*, *23*(6), 703–713. https://doi.org/10.1038/nm.4333

Zhang, M., Fritsche, J., Roszik, J., Williams, L. J., Peng, X., Chiu, Y., Tsou, C. C., Hoffgaard, F., Goldfinger, V., Schoor, O., Talukder, A., Forget, M. A., Haymaker, C., Bernatchez, C., Han, L., Tsang, Y. H., Kong, K., Xu, X., Scott, K. L., … Hwu, P. (2018). RNA editing derived epitopes function as cancer antigens to elicit immune responses. *Nature Communications*, *9*(1), 1–10. https://doi.org/10.1038/s41467-018-06405-9

Zheng, L., Qin, S., Si, W., Wang, A., Xing, B., Gao, R., Ren, X., Wang, L., Wu, X., Zhang, J., Wu, N., Zhang, N., Zheng, H., Ouyang, H., Chen, K., Bu, Z., Hu, X., Ji, J., & Zhang, Z. (2021). Pan-cancer single-cell landscape of tumor-infiltrating T cells. *Science*, *374*(6574). https://doi.org/10.1126/science.abe6474

Zhou, C., Wei, Z., Zhang, L., Yang, Z., & Liu, Q. (2020). Systematically Characterizing A-to-I RNA Editing Neoantigens in Cancer. *Frontiers in Oncology*, *10*, 2753. https://doi.org/10.3389/fonc.2020.593989

Zitvogel, L., Perreault, C., Finn, O. J., & Kroemer, G. (2021). Beneficial autoimmunity improves cancer prognosis. *Nature Reviews Clinical Oncology 2021 18:9*, *18*(9), 591–602. https://doi.org/10.1038/s41571-021-00508-x

6. Appendix

6.1 Detailed patient information

Detailed information for each tumor sample in the ImmuNEO cohort including entity, metastatic site (or primary), stage at admission and the primary sampling cohort. Core samples utilized for statistical analysis (see “subset” column) are marked. Tumor samples employed for immune phenotyping of the tumor microenvironment (TME) through flow cytometry analysis and bulk RNA sequencing (RNA-seq) of sorted CD8$^+$ T cells are indicated. Samples subjected to whole exome sequencing (WES) and tumor RNA-seq are labelled respectively, those samples analysed via whole genome sequencing (WGS) instead of WES are marked with an asterisk. The survival status along with the survival times in months are displayed for various periods: since initial diagnosis (ID), diagnosis of metastatic disease (MD) and admission to MASTER/tumor resection (MASTER). The time difference between MD and MASTER is provided in months. Additionally, details about patients receiving immune checkpoint blockade before and/or after study admission, along with their respective responses (no response = 0, mixed response = 1 and good response = 2), are included. Ca, carcinoma; DSRCT, desmoplastic small round cell tumor; MPNST, malignant peripheral nerve sheath tumor; GIST, gastrointestinal stromal tumor; LN, lymph node; IN, ImmuNEO; MS, mass spectometry; WES, whole exome sequencing; WGS, whole genome sequencing; RNA-seq, RNA sequencing; IME, immune microenvironment.

General information					Subset	IME analysis		MS	Mutation Analysis		Survival data [months]					Immunotherapy Checkpoint Therapy					
Patient ID:	Tumor entity	Metastatic site	Staging at admission	Cohort	Core Samples	Pheno-typing	Sort & RNAseq	MS	WES (*WGS)	RNAseq tumor	Survival status	since ID	since MD	since MASTER	MD - MASTER	Received general	Response	Received prior admission	Response prior	Received post admission	Response post
ImmuNEO-1.1	Thymoma	lung	Stage Ivb	MASTER	√	x	x	√	√	√	alive	51	51	46	5	Yes	2	x	-	Yes	2
ImmuNEO-1.2		lung pericardium		-	x	x	x	x	x	x											
ImmuNEO-2	Mamma-Ca	primary	Stage IIb	MASTER	√	x	x	√	√	√	alive	41	41	37	4	x	-	x	-	x	-
ImmuNEO-3	Sarcoma (DSRCT)	primary	Stage IV	MASTER	√	x	x	√	√	√	deceased	29	27	23	4	x	-	x	-	x	-
ImmuNEO-4	Renal-cell-Ca	LN	Stage IV	MASTER	√	√	x	√	√	√	deceased	35	35	12	23	Yes	2	Yes	2	Yes	1
ImmuNEO-5	Leiomyosarcoma	lung	Stage IV	MASTER	√	√	x	√	√	√	deceased	17	17	4	13	x	-	x	-	x	-
ImmuNEO-8	Ovarian-Ca (neuroendocrine)	subcutaneous hypogastrium	Stage IV	MASTER	√	√	x	√	√	√	deceased	65	65	21	44	Yes	0	x	-	Yes	0
ImmuNEO-9	Thyroid-Ca	LN	Stage IV	MASTER	√	x	x	√	√	√	alive	39	39	39	0	x	-	x	-	x	-
ImmuNEO-11.1	Endometrium-Ca	primary	*n.a.*	MASTER	x	√	√	√	√	x	alive	63	63	32	31	x	-	x	-	x	-
ImmuNEO-11.2	Pancreas-Ca	LN	Stage IV	MASTER	√	√	√	√	√	√											
ImmuNEO-13	Testicle-Ca	LN	Stage IIIb	MASTER	√	x	x	√	√	√	deceased	21	21	10	10	x	-	x	-	x	-
ImmuNEO-14	Melanoma	subcutaneous abdominal wall	Stage IV	MASTER	√	x	x	√	√	x	deceased	127	11	6	5	Yes	0	Yes	0	Yes	0
ImmuNEO-15	Testicle-Ca	lung	Stage IIICA	MASTER	√	√	√	√	√	√	alive	128	124	35	89	Yes	0	x	-	Yes	0
ImmuNEO-16	Adeno-Ca (Gl. sublingualis)	primary	Stage IV	MASTER	√	x	x	√	√	x	deceased	24	24	17	7	Yes	0	x	-	Yes	0
ImmuNEO-17.1	Melanoma	LN	Stage IV	IN Plus	x	x	x	√	x	x	alive	213	68	31	37	Yes	1	Yes	1	x	-
ImmuNEO-17.2		LN		IN Plus	√	√	x	√	√	√											
ImmuNEO-17.3		LN		IN Plus	x	x	x	√	√	√											
ImmuNEO-18	Mamma-Ca	ovar	Stage IV	IN Plus	√	√	x	√	√	√	deceased	62	62	22	40	x	-	x	-	x	-
ImmuNEO-19.1	Melanoma	LN colon	Stage IV	IN Plus	x	√	√	√	√	√	alive	30	30	29	1	Yes	2	x	-	Yes	2
ImmuNEO-19.2		colon		IN Plus	x	√	√	√	√	√											
ImmuNEO-19.3		colon		IN Plus	x	√	√	√	x	x											
ImmuNEO-19.4		liver		IN Plus	√	√	√	√	√	√											
ImmuNEO-20	Testicle-Ca	LN	Stage IIIB	MASTER	√	√	√	√	√	x	deceased	90	15	8	7	x	-	x	-	x	-
ImmuNEO-22	Melanoma	abdominal wall	Stage IV	MASTER	√	x	x	√	√ *	√	alive	67	43	31	12	Yes	1	Yes	1	Yes	1
ImmuNEO-23.1	Sarcoma (MPNST)	LN	Stage IV	IN Plus	x	√	x	√	√	√	deceased	12	12	8	4	x	-	x	-	x	-
ImmuNEO-23.2		primary (thorax)		MASTER	√	√	√	√	√	√											
ImmuNEO-24.1	Adrenocortical-Ca	liver	Stage IV	IN Plus	√	√	√	√	√	√	alive	33	33	32	1	x	-	x	-	x	-
ImmuNEO-24.2		primary (kidney)		MASTER	x	√	√	√	√	√											
ImmuNEO-25	Sarcoma (GIST)	primary (intestine)	Stage IV	MASTER	√	√	√	√	√	x	alive	139	139	15	124	x	-	x	-	x	-
ImmuNEO-26	Adeno-Ca (mucoepidermoid)	primary	Stage IVA	MASTER	√	x	x	√	√	√	deceased	9	9	5	3	Yes	1	x	-	Yes	1
ImmuNEO-27.1	Fibrosarcoma (epitheloid)	primary	Stage IV	IN Plus	x	x	x	√	√	√	alive	35	27	26	1	x	-	x	-	x	-
ImmuNEO-27.2		lung		MASTER	√	x	x	√	√	√											
ImmuNEO-28	Clear cell sarcoma	primary	Stage IV	MASTER	√	√	x	√	√	√	alive	27	27	25	2	x	-	x	-	x	-
ImmuNEO-30	Synovial sarcoma	primary	Stage IV	MASTER	√	x	x	√	√ *	√	alive	34	34	26	8	x	-	x	-	x	-
ImmuNEO-31	Rhabdomyosarcoma	primary	Stage IV	MASTER	√	x	x	√	√	x	deceased	15	15	14	1	x	-	x	-	x	-
ImmuNEO-32	Osteosarcoma	brain	Stage V-VI	MASTER	√	√	√	√	√	√	deceased	17	2	2	0	x	-	x	-	x	-
ImmuNEO-33	atypical carcinoid of the lung	asubcut. thorax	Stage IV	MASTER	√	x	x	√	√	√	deceased	18	18	5	13	x	-	x	-	x	-
ImmuNEO-34	Adeno-Ca (mucinous, appendix)	primary	Stage IV	MASTER	√	x	x	√	√	x	deceased	6	6	3	3	x	-	x	-	x	-
ImmuNEO-35	Fibrosarcoma (prostate)	*n.a.*	Stage IV	MASTER	√	√	x	√	√	√	alive	34	13	9	4	x	-	x	-	x	-
ImmuNEO-36	Adeno-Ca (Barret-Ca)	LN	AEG I; G3	MASTER	√	x	x	√	√	√	deceased	6	6	6	0	Yes	0	x	-	Yes	0
ImmuNEO-37	Adeno-Ca (appendix)	primary	Stage IVc	MASTER	√	√	x	√	√	√	deceased	16	16	4	12	x	-	x	-	x	-
ImmuNEO-38	Sarcoma (MPNST)	colon	Stage IV	MASTER	√	√	x	√	√	√	alive	64	34	13	21	x	-	x	-	x	-

6.2 Overview of applied therapies to patients within the ImmuNEO cohort

Details regarding the therapies administered to each ImmuNEO patient both before and after tumor resection are provided. 1 = therapy applied, 0 = therapy not applied. IN, ImmuNEO; OP, operation; Chemo, chemotherapy.

	Therapy prior to sample extraction					Therapy after sample extraction				
ImmuNEO-ID	OP	x-Ray	Chemo	Targeted Therapy	Checkpoint Therapy	OP	x-Ray	Chemo	Targeted Therapy	Checkpoint Therapy
IN-01	0	0	1	0	0	0	0	0	0	1
IN-02	1	1	1	0	0	0	0	0	1	0
IN-03	1	0	1	1	0	0	0	1	1	0
IN-04	0	0	0	1	1	0	1	0	1	1
IN-05	1	0	1	0	0	0	0	0	0	0
IN-08	1	1	1	0	0	0	1	1	0	1
IN-09	0	0	0	0	0	0	1	0	1	0
IN-11	1	0	0	0	0	1	0	0	0	0
IN-13	1	0	1	0	0	0	0	1	0	0
IN-14	1	1	0	1	1	0	0	1	0	1
IN-15	1	1	1	0	0	1	0	1	1	1
IN-16	0	0	1	0	0	0	1	0	0	1
IN-17	0	1	0	0	1	0	1	0	0	0
IN-18	0	1	1	0	0	0	1	0	0	0
IN-19	0	0	0	0	0	1	0	0	0	1
IN-20	1	0	1	0	0	0	1	1	0	0
IN-22	1	1	0	0	1	0	0	1	1	1
IN-23	1	0	1	0	0	1	1	1	0	0
IN-24	0	0	1	0	0	1	0	1	1	0
IN-25	1	0	0	1	0	0	0	0	0	0
IN-26	1	1	1	0	0	0	0	1	0	1
IN-27	1	0	0	0	0	1	0	1	1	0
IN-28	0	0	0	0	0	0	0	0	0	0
IN-30	1	1	1	0	0	1	0	1	0	0
IN-31	0	0	1	0	0	0	1	1	0	0
IN-32	1	0	1	0	0	0	0	1	0	0
IN-33	1	1	1	0	0	0	0	0	0	0
IN-34	0	0	1	0	0	0	0	0	0	0
IN-35	1	1	1	1	0	0	0	1	1	0
IN-36	1	0	0	0	0	0	0	1	0	1
IN-37	1	0	1	1	0	1	0	1	1	0
IN-38	1	1	0	0	0	0	0	0	0	0

6.3 Neoantigen candidate overview

By combining genomic mutational data with MS-based immunopeptidomic data for each patient sample, neoantigen candidates were identified. pFIND (v3.1.5) was used at 5% FDR on spectral level for the identification of non-wild type 8-15mer neoantigen candidates. The machine learning tool Prosit was additionally integrated to rescore and rematch the peptide spectra using unfiltered pFIND data as input. n = 39 tumor samples from n = 32 patients were analysed in total; n = 27 tumor samples from n = 24 patients harboured n = 90 neoantigen candidates. Using netMHC4.0 and MHCFlurry, binding predictions for each peptide towards the patients six HLA class I alleles (see Table 6) was performed and for each algorithm the best binding allele by affinity and by rank are shown. Mutated amino acids are marked with two asterisks within the sequence and the variant location is annotated in 5' to 3' direction. Additional information for each peptide and variant is given such as variant frequency within the tumor, the coverage of the variant on DNA and RNA level, the GTEx prevalence of the variant, peptide verification data (SA and RT errors) as well as the immunogenicity of the peptide defined by acDC (see 3.6.1). a.a, amino acid; Alt, alternative; BA, binding affinity; Chrom, chromosome; del, deletion; dup, duplication; HLA, human leukocyte antigen; ins, insertion; MS, mass spectrometry; n.a./NA, not applicable; nM nanomolar; Pos, position; Ref, reference; RT, retention time; SA, spectral contrast angle; Seq, sequence; T, tumor; VF, variant frequency.

Patient Tumor	Seq ID	Peptide length	Sequence	Ref_Alt	a.a Ref_Alt	Chrom:Pos	Gene	MS tool	Mutation calling algorithm	MHCFlurry_BA [nM, HLA allele]	MHCFlurry_rank [%rank, HLA allele]	netMHC_BA [nM, HLA allele]	netMHC_rank [%rank, HLA allele]	VF_DNA	VF_RNA	Coverage _DNA	Coverage_ RNA	GTEx hits (%)	Best SA score	Best RT error abs.	Immun ogenic	Validated
IN_01_T1	IN_01_A	9	ALSGHLET*L*	G_T	Val98Leu	9:33624565	ANXA2P2	pFind + PROSIT	StrelkaRNA	14.45 ; HLA-A0201	0.05 ; HLA-A0201	44.3 ; HLA-A0201	0.6 ; HLA-A0201	*n.a.*	0.44	249	347	13.224	0.89	4.29	yes	not
IN_01_T1	IN_01_B	11	KGDSPQVKLKY	G_A	intronic	10:58394357	TFAM	pFind + PROSIT	StrelkaRNA	4179.49 ; HLA-C0501	3.88 ; HLA-C0501	35793.9 ; HLA-B4402	17 ; HLA-C0501	*n.a.*	0.67	20	60	0	0.77	3.54	yes	fully
IN_01_T1	IN_01_C	9	GHPSGARAM	G_C	intronic	19:16127527	RAB8A	pFind	StrelkaRNA	56.88 ; HLA-C0704	0.6 ; HLA-C0704	28556.2 ; HLA-B4402	15 ; HLA-C0501	*n.a.*	1	203	6	0.273	0.09	16.05	yes	not
IN_01_T1	IN_01_D	9	KELCKQIQL	G_T	intronic	7:30632246	GARS	PROSIT	StrelkaRNA	52.04 ; HLA-C0704	0.3 ; HLA-B4402	350.7 ; HLA-B4402	0.5 ; HLA-B4402	*n.a.*	0.35	184	31	0.204	0.70	31.62	no	partly
IN_02_T1	IN_02_A	9	TGGQKYRTK	A_G	intronic	9:72360223	ZFAND5	pFind	StrelkaRNA	2176.27 ; HLA-A1101	2.49 ; HLA-A1101	12238.1 ; HLA-A1101	10 ; HLA-A1101	*n.a.*	0.43	55	883	89.035	0.14	3.43	*n.a.*	not
IN_03_T1	IN_03_A	10	A*A*SASRVQVI	A_G	Thr35Ala	5:139282911	SNHG4	pFind	StrelkaRNA	1747.46 ; HLA-C0602	1.42 ; HLA-B5101	18274.8 ; HLA-B5101	7 ; HLA-B5101	*n.a.*	0.10	1	50	4.518	0.31	3.29	no	not
IN_03_T1	IN_03_C	8	ESKDFCVM	T_A	intronic	5:77693264	TBCA	pFind	StrelkaRNA	12946.13 ; HLA-B3701	8.14 ; HLA-B5101	30879.4 ; HLA-A0101	8 ; HLA-B3701	*n.a.*	0.24	212	51	27.130	0.94	13.54	no	not
IN_03_T1	IN_03_D	9	G*S*HDQAMHF	A_G	Asn337Ser	1:108903182	GPSM2	pFind + PROSIT	StrelkaRNA	244.33 ; HLA-C0602	1.28 ; HLA-C0602	3542.7 ; HLA-A2403	3 ; HLA-A2403	*n.a.*	0.03	116	219	0	0.72	21.95	no	not
IN_03_T1	IN_03_E	9	TDGGGRAKL	G_A	intronic	3:5170502	ARL8B	pFind + PROSIT	StrelkaRNA	64.45 ; HLA-B3701	0.65 ; HLA-B3701	16221.8 ; HLA-C1402	9 ; HLA-C1402	*n.a.*	0.08	62	75	0	0.72	39.71	no	partly
IN_03_T1	IN_03_F	9	TFQ*K*KTKEM	dupAGA	dupl K5	9:65649319	FRG1KP	pFind + PROSIT	StrelkaRNA	19.24 ; HLA-C1402	0.02 ; HLA-C1402	66.7 ; HLA-C1402	0.25 ; HLA-C1402	*n.a.*	0.50	2	8	0	0.94	1.08	no	fully
IN_04_T1	IN_04_A	8	AGVVLGGL	G_A	intronic	19:38380898	PSMD8	pFind	StrelkaRNA	13993.24 ; HLA-C1203	14.07 ; HLA-C1203	34748.4 ; HLA-B1501	65 ; HLA-B1501	*n.a.*	0.71	73	7	22.875	0.88	6.92	yes	not
IN_04_T1	IN_04_B	9	FLLLLLKNF	G_A	intronic	2:197416740	SF3B1	pFind	StrelkaRNA	565.32 ; HLA-B1501	0.91 ; HLA-A2301	1604 ; HLA-A2301	2 ; HLA-A2301	*n.a.*	0.83	94	6	0	0.58	42.06	no	not
IN_04_T1	IN_04_C	9	GL*A*ATFASL	A_G	Thr21Ala	1:32234241	MTMR9LP	pFind	StrelkaRNA	223.81 ; HLA-B1501	0.98 ; HLA-B1501	35.1 ; HLA-B1501	0.4 ; HLA-B1501	*n.a.*	0.68	5	19	46.791	0.71	21.60	no	not
IN_04_T1	IN_04_D	10	KTKEM*S*NNVK	A_C	STOP11Ser	20:29100437	FRG1DP	pFind	StrelkaRNA	4355.18 ; HLA-C0602	7.9 ; HLA-C0602	20860.1 ; HLA-B1501	29 ; HLA-B1501	*n.a.*	0.83	2	12	22.699	0.72	1.07	no	not
IN_04_T1	IN_04_E	8	LGG*T*GASF	G_A	Ala42Thr	3:161429566	AC112491.1	pFind	StrelkaRNA	341.35 ; HLA-C1203	2.5 ; HLA-C1203	2561.8 ; HLA-B1501	6 ; HLA-B1501	*n.a.*	0.005	164	1793	0.506	0.39	0.81	no	not
IN_04_T1	IN_04_F	9	NTLMSLSDM	G_A	intronic	14:50440012	MAP4K5	pFind	StrelkaRNA	1398.2 ; HLA-C1203	4.35 ; HLA-C1203	13721 ; HLA-B1501	7 ; HLA-C0602	*n.a.*	0.98	13	1675	3.944	0.28	16.53	no	not
IN_04_T1	IN_04_G	8	SYLSNISY	G_A	intronic	8:61637946	ASPH	pFind + PROSIT	StrelkaRNA	7562.4 ; HLA-C0602	3.22 ; HLA-A2301	4025 ; HLA-B1501	5.5 ; HLA-A2301	*n.a.*	0.78	33	82	0	0.89	38.04	no	not
IN_04_T1	IN_04_H	8	TSLA*A*NTF	G_C	Gly85Ala	6:149978178	BTF3P10	pFind	StrelkaRNA	204.51 ; HLA-C1203	1.97 ; HLA-C1203	5628.3 ; HLA-A2301	4.5 ; HLA-A2301	*n.a.*	1	37	6	0.312	0.79	31.52	no	not
IN_04_T1	IN_04_I	9	*T*VHSTSIAF	T_C	Ile9Thr	9:128342391	TMSB4XP4	pFind + PROSIT	StrelkaRNA	26.03 ; HLA-B1501	0.06 ; HLA-B1501	28.5 ; HLA-B1501	0.3 ; HLA-B1501	*n.a.*	0.17	17	24	4.966	0.90	2.69	yes	fully
IN_04_T1	IN_04_J	9	GHGQPWNSL	G_C	intronic	12:57156284	LRP1	pFind	StrelkaRNA	39.29 ; HLA-B3801	0.1 ; HLA-B3801	791.2 ; HLA-B3801	0.25 ; HLA-B3801	*n.a.*	0.60	63	5	0	0.09	46.85	yes	not
IN_04_T1	IN_04_K	9	**HAGAALHLH	insT	Leu294 ins	6:31900430	ZBTB12	pFind	StrelkaRNA	251.26 ; HLA-C1203	2.18 ; HLA-C1203	18117.7 ; HLA-C1203	11 ; HLA-C0602	*n.a.*	0.11	4	27	0	0.35	39.50	yes	not
IN_04_T1	IN_04_L	10	KLQNA*S*KKLF	C_T	Pro175Ser	18:6241387	L3MBTL4	pFind	StrelkaRNA	156.42 ; HLA-B1501	0.82 ; HLA-B1501	186.1 ; HLA-B1501	1.2 ; HLA-B1501	*n.a.*	0.33	41	18	0	0.84	2.51	no	fully
IN_04_T1	IN_04_M	8	KSAGI*A*GL	A_G	Thr14Ala	17:81880732	AC145207.5	pFind	StrelkaRNA	1496.84 ; HLA-C1203	4.48 ; HLA-C1203	19304.7 ; HLA-B1501	21 ; HLA-B3801	*n.a.*	0.50	5	4	0.282	0.50	24.48	yes	not
IN_05_T1	IN_05_A	9	DIFSRISQ*R*	A_G	Gln54Arg	9:18481 + 1:188891	WASHC1;FO538757.1	PROSIT	StrelkaRNA	19.92 ; HLA-A6601	0.01 ; HLA-A6601	6727.8 ; HLA-A0301	4 ; HLA-A6601	*n.a.*	0.55	1 + 3	800+1146	4.888	0.92	1.89	yes	fully
IN_05_T1	IN_05_B	9	E*T*NKSLLKR	T_C	Met15Thr	21:9327285	CR381653.1	pFind + PROSIT	StrelkaRNA	20.12 ; HLA-A6601	0.01 , HLA-A6601	3934.4 ; HLA-A6601	0.4 ; HLA-A6601	*n.a.*	0.94	114	71	4.528	0.82	3.28	*n.a.*	fully
IN_05_T1	IN_05_C	9	DLLEPG*G*QR	A_G	Arg20Gly	19:20207426	AC011447.6	PROSIT	StrelkaRNA	25.16 ; HLA-A6601	0.05 ; HLA-A6601	19618.6 ; HLA-A6601	15 ; HLA-A6601	*n.a.*	1	2	5	0	0.69	19.47	*n.a.*	not
IN_05_T1	IN_05_D	9	SL*G*AGRWRL	A_G	Glu45Gly	12:22460675	AC053513.1	pFind	StrelkaRNA	1159.24; HLA-C1703	3.47 ; HLA-C1703	13593.6 ; HLA-A6601	7 ; HLA-A6601	*n.a.*	0.13	3	23	0.575	0.27	46.21	*n.a.*	not
IN_08_T1	IN_08_A	9	LSELDVSVR	G_C	intronic	1:26942870	NUDC	pFind	StrelkaRNA	3433.55 ; HLA-B2705	4.01 ; HLA-A0301	23185.7 ; HLA-A0301	23 ; HLA-A0301	*n.a.*	0.19	71	115	0.068	0.24	14.18	no	not
IN_08_T1	IN_08_B	9	PQESAPAAL	T_A	intronic	2:36356625	CRIM1	pFind	StrelkaRNA	915.8 ; HLA-B0702	1.44 ; HLA-B0702	10007.6 ; HLA-C0702	3.5 ; HLA-C0702	*n.a.*	0.67	362	3	0	0.67	0.97	no	partly
IN_08_T1	IN_08_C	8	APVLKS*A*R	C_G	Pro92Ala	6:30006556	HLA-J	pFind	StrelkaRNA	7300.2 ; HLA-B0702	3.2 ; HLA-B0702	14416.5 ; HLA-B0702	10 ; HLA-B0702	*n.a.*	0.05	312	113	0	0.62	43.66	*n.a.*	not
IN_08_T1	IN_08_D	9	GLEPGKCSP	A_T	intronic	19:19121821	TMEM161A	pFind	StrelkaRNA	9166.88 ; HLA-B2705	11.07 , HLA-B0702	30662 ; HLA-A0301	25 ; HLA-C0702	*n.a.*	0.25	221	12	0.282	0.14	1.78	*n.a.*	not
IN_08_T1	IN_08_E	9	GPLGPR*G*SI	T_G	Val??Gly	2:189058847	COL5A2	pFind	StrelkaRNA	24.24 ; HLA-B0702	0.03 ; HLA-B0702	8.6 ; HLA-B0702	0.03 ; HLA-B0702	*n.a.*	0.5	252	6	0.029	0.43	10.60	*n.a.*	not
IN_08_T1	IN_08_F	9	NRITEVSAK	G_A	intronic	8:91063747	OTUD6B-AS1	pFind + PROSIT	StrelkaRNA	134.35 ; HLA-B2705	0.39 ; HLA-B2705	1011.4 ; HLA-B2705	2.5 ; HLA-B2705	*n.a.*	0.33	1	9	0	0.95	4.20	*n.a.*	fully
IN_08_T1	IN_08_G	14	SAGAAAQGRAGGAP	A_T	intronic	11:78280903	GAB2	PROSIT	StrelkaRNA	2139.37 ; HLA-B2705	4.7 ; HLA-B2705	35732 ; HLA-B0702	55 ; HLA-B0702	*n.a.*	0.33	193	6	0	0.56	5.13	*n.a.*	partly
IN_08_T1	IN_08_H	10	TQAL*V*LAPTQ	C_G	Leu89Val	12:53153584	EIF4A1P4	pFind	StrelkaRNA	13177.86 ; HLA-B2705	21 ; HLA-B2705	20958.1 ; HLA-B2705	17 ; HLA-A2601	*n.a.*	1	31	10	0.419	0.43	27.30	*n.a.*	not
IN_11_T1 + T2	IN_11_A	8	SAAEL*H*HV	G_A	Arg107His	19:46608880	CALM3	pFind	Mutect2	44.11 ; HLA-B5201	0.24 ; HLA-B5201	11800.7 ; HLA-A0201	17 ; HLA-A0201	0.08+0.07	*n.a.*	52 + 44	*n.a.* +3105	0	0.45	4.15	no	not
IN_11_T2	IN_11_B	9	GGITAVT*L*N	G_C	Val9Leu	5:123402142	KRT8P33	pFind + PROSIT	StrelkaRNA	4098.52 ; HLA-B2705	7.93 ; HLA-B2705	29854.7 ; HLA-B2705	39 ; HLA-B2705	*n.a.*	0.33	17	9	0	0.65	10.69	yes	not
IN_11_T1	IN_11_C	9	**RGISWRSHL	dupC	Leu894 ins	10:133466247	SCART1	pFind	Mutect2	254.38 ; HLA-C0102	1.25 ; HLA-B2705	2309.8 ; HLA-B2705	3.5 ; HLA-B2705	0.22		50	*n.a.*	0	0.42	26.74	no	not
IN_11_T2	IN_11_D	11	S*R*SVAQAGVQR	T_C	Cys4Arg	8:94792842	AP003692.1	PROSIT	StrelkaRNA	154.19 ; HLA-B2705	0.47 ; HLA-B2705	5974 ; HLA-B2705	6.5 ; HLA-B2705	*n.a.*	0.26	17	35	77.028	0.58	0.73	yes	not
IN_11_T2	IN_11_E	8	**VAAGPGAV	delGT	Leu32 del	21:32771571	PAXBP1	pFind	StrelkaRNA	83.67 ; HLA-C1202	1.43 ; HLA-C1202	27480.4 ; HLA-A0201	41 ; HLA-A0201	*n.a.*	0.08	99	39	0	0.28	39.83	no	not
IN_13_T1	IN_13_A	9	KLPTLPKKY	G_A	intronic	4:25388716	ANAPC4	pFind	StrelkaRNA	348.48 ; HLA-C0701	1.58 ; HLA-C0701	16936.9 ; HLA-A1101	14 ; HLA-A1101	*n.a.*	1	60	4	0	0.64	4.30	no	not
IN_13_T1	IN_13_B	8	LFKNLTIL	G_T	intronic	2:149571084	MMADHC	PROSIT	StrelkaRNA	931.22 ; HLA-B0801	1.48 ; HLA-B0801	2731.8 ; HLA-B0801	3 ; HLA-B0801	*n.a.*	0.55	59	20	0	0.73	34.52	no	not
IN_15_T1	IN_15_A	8	ICTT*S*VSK	C_T	Pro57Ser	13:68985414	ZDHHC20P4	pFind	StrelkaRNA	7897.05 ; HLA-C0602	5.86 ; HLA-A0301	7234.4 ; HLA-A0301	6.5 ; HLA-A0301	*n.a.*	1	3	5	0.019	0.21	3.49	no	not
IN_15_T1	IN_15_B	9	LRAVTL*I*AK	C_T	Thr21Ile	15:74479279	AC012435.1	pFind	StrelkaRNA	1574.26 ; HLA-C0602	2.9 ; HLA-A0301	8741.2 ; HLA-A0301	7.5 ; HLA-A0301	*n.a.*	0.38	2	8	0.049	0.62	36.95	no	not
IN_17_T2	IN_17_A	8	MQSRLTA*A*	A_G	Thr21Ala	19:40358347	AC118344.2	pFind+PROSIT	StrelkaRNA	353.6 ; HLA-B3906	1.1 ; HLA-B3906	14395.8 ; HLA-A0201	9.5 ; HLA-B1402	*n.a.*	0.29	3	7	1.402	0.71	2.78	*n.a.*	fully
IN_17_T3	IN_17_B	8	*A*GLSHHAL	A_G	Thr14Ala	17:81880732	AC145207.5	pFind	StrelkaRNA	89.14 ; HLA-B1402	0.44 ; HLA-B1402	23137.9 ; HLA-B1402	15 ; HLA-B1402	*n.a.*	0.22	5	9	0.282	0.08	10.10	*n.a.*	not
IN_18_T1	IN_18_A	8	MRL*W*SQLL	A_G	STOP4Trp	7:128526780	AC090114.2	pFind	StrelkaRNA	576.62 ; HLA-C0602	2.29 ; HLA-C0602	256.8 ; HLA-C0602	0.15 ; HLA-C0602	*n.a.*	0.13	8	83	15.425	0.57	71.66	no	not

Patient Tumor	Seq ID	Peptide length	Sequence	Ref_Alt	a.a Ref_Alt	Chrom:Pos	Gene	MS tool	Mutation calling algorithm	MHCFlurry_BA_best [nM, HLA allele]	MHCFlurry_rank_best [%rank, HLA allele]	netMHC_BA_best [nM, HLA allele]	netMHC_rank_best [%rank, HLA allele]	VF_DNA	VF_RNA	Coverage_DNA	Coverage_RNA	GTEx hits (%)	Best SA score	Best RT error abs.	Immunogenic	Validated
IN_19_T2+T4	IN_19_A	9	GRPGTRPAL	G_T	intronic	12:119990298	BICDL1	pFind	StrelkaRNA	846.32 ; HLA-C1601	1.9 ; HLA-C0401	11371.1 ; HLA-C0401	3.5 ; HLA-C0401	*n.a.*	1	139+122	1	0.565	0.75	1.70	yes	fully
IN_19_T4	IN_19_B	9	SES*N*VDRLM	G_A	Ser356Asn	7:130561094	COPG2	pFind + PROSIT	StrelkaRNA	30.64 ; HLA-B4403	0.1 ; HLA-B4403	343.8 ; HLA-B4403	0.5 ; HLA-B4403	*n.a.*	0.50	0	336	0	0.83	1.71	yes	fully
IN_19_T2	IN_19_C	10	ST*L*VLDEFKR	T_C	Phe266Leu	7:133035059	AC008038.1	pFind + PROSIT	StrelkaRNA	3364.78 ; HLA-C1601	6.13 ; HLA-A0101	26273.5 ; HLA-A2902	28 ; HLA-A2902	*n.a.*	0.001	145	9932	0.136	0.34	48.44	yes	not
IN_19_T2+T4	IN_19_D	8	**VASISLTK	delT	Phe11 del	14:49619072	RPL36AL	pFind + PROSIT	StrelkaRNA+Mutect2	493.32 ; HLA-C1601	3.8 ; HLA-C1601	35248.2 ; HLA-A0101	60 ; HLA-A2902	0.20	0.15+0.17	230+208	2580+3554	0	0.90	0.11	no	fully
IN_19_T4	IN_19_E	14	*G*SLNGGKPFLQAFY	A_G	Glu124Gly	15:58010771	ALDH1A2	PROSIT	StrelkaRNA	10183.47 ; HLA-A2902	3.22 ; HLA-A0101	752.8 ; HLA-A2902	0.9 ; HLA-A0101	*n.a.*	0.27	80	11	0	0.63	8.02	yes	partly
IN_19_T2	IN_19_F	9	KKY*W*VGAKL	A_G	STOP4Trp	11:113368805	AP002840.2	PROSIT	StrelkaRNA	17705.52 ; HLA-C1601	7.37 ; HLA-B4403	5104.2 ; HLA-C0401	0.4 ; HLA-C0401	*n.a.*	0.50	0	6	15.230	0.61	36.70	yes	not
IN_19_T4	IN_19_G	8	KVGSLAG*F*	C_T	Ser722Phe	10:99364991	CNNM1	pFind	StrelkaRNA	10247.14 ; HLA-C1601	14.06 ; HLA-C1601	27203.2 ; HLA-A2902	24 ; HLA-B4403	*n.a.*	0.33	197	6	0	0.43	46.51	no	not
IN_19_T2+T4	IN_19_H	9	MPEHQSTAL	T_A	intronic	6:31944855	C2;AL645922.1	pFind + PROSIT	StrelkaRNA	27.53 ; HLA-B3502	0.01 ; HLA-B3502	6656.6 ; HLA-C0401	0.9 ; HLA-C0401	*n.a.*	0.1+0.05	222+229	30 + 38	0.185	0.86	1.04	yes	fully
IN_19_T2	IN_19_I	9	*R*RLQRDKIA	A_G	Gln286Arg	17:82483251	NARF	pFind	StrelkaRNA	29688.49 ; HLA-C1601	15.1 ; HLA-B4403	26222.4 ; HLA-C0401	19 ; HLA-C0401	*n.a.*	0.02	23	66	15.211	0.23	6.75	yes	not
IN_22_T1	IN_22_A	9	PPSEAQP*L*P	A_T	Gln197Leu	15:45403019	SPATA5L1	pFind	StrelkaRNA	24670.46 ; HLA-C0501	18.69 ; HLA-C0501	36553.2 ; HLA-A2902	33 ; HLA-C0501	*n.a.*	0.22	62	18	0	0.42	21.69	yes	not
IN_23_T1	IN_23_A	14	ASASQSA*G*IIGMSH	A_G	Arg39Gly	19:16635111	AC024075.2	pFind	StrelkaRNA	17820.35 ; HLA-C0702	7.73 ; HLA-A1101	15379.7 ; HLA-A1101	12 ; HLA-A1101	*n.a.*	0.25	3	40	56.666	*n.a.*	*n.a.*	no	not
IN_23_T1	IN_23_B	10	GAPAPVMVEK	G_T	intronic	2:237376771	COL6A3	PROSIT	StrelkaRNA	39.54 ; HLA-A1101	0.29 ; HLA-A1101	49.1 ; HLA-A1101	0.4 ; HLA-A1101	*n.a.*	0.25	182	8	0	0.53	8.07	no	partly
IN_24_T2	IN_24_A	10	S*R*VVGITGVP	T_C	STOP49Arg	15:84641638	SCAND2P	pFind	StrelkaRNA	11360.23 ; HLA-B2704	6.51 ; HLA-B2704	18022.9 ; HLA-A0206	31 ; HLA-A0206	*n.a.*	0.04	5	52	18.259	0.08	37.27	no	not
IN_24_T2	IN_24_B	8	LPIYGRAR	G_C	intronic	3:134952053	EPHB1	PROSIT	StrelkaRNA	16395.42 ; HLA-C1202	15.35 ; HLA-A1101	37530.3 ; HLA-A1101	60 ; HLA-A1101	*n.a.*	0.38	70	8	0	0.85	19.08	no	partly
IN_24_T2	IN_24_C	12	STMVKGRQTTTK	T_G	intronic	12:21637069	LDHB	PROSIT	StrelkaRNA	535.99 ; HLA-A1101	1.54 ; HLA-A1101	50.2 ; HLA-A1101	0.4 ; HLA-A1101	*n.a.*	1	3	2	0.019	0.46	5.43	no	not
IN_27_T2	IN_27_A	9	EGVAGPHSR	G_A	intronic	9:93075783	SUSD3	pFind	StrelkaRNA	29.42 ; HLA-A3301	0.06 ; HLA-A3301	730.7 ; HLA-A3301	1.5 ; HLA-A3301	*n.a.*	0.67	34	3	0	0.91	3.68	no	fully
IN_28_T1	IN_28_A	11	RVWD*V*SGLRKK	A_G	Ile164Val	1:160332454	COPA	pFind	StrelkaRNA	26.84 ; HLA-A3001	0.03 ; HLA-A3001	20287.6 ; HLA-A3001	18 ; HLA-C0401	*n.a.*	0.43	160	478	84.517	0.87	2.59	no	not
IN_28_T1	IN_28_B	9	SP*R*QPPLLL	A_G	Gln135Arg	7:39950745	CDK13	pFind + PROSIT	StrelkaRNA	18.83 ; HLA-B0702	0 ; HLA-B0702	9.7 ; HLA-B0702	0.04 ; HLA-B0702	*n.a.*	0.56	40	9	16.779	0.85	3.41	no	not
IN_28_T1	IN_28_C	9	VIHPP*R*PPK	A_G	Gln18Arg	1:20649530	PINK1-AS	pFind + PROSIT	StrelkaRNA	21.19 ; HLA-A3001	0.01 ; HLA-A3001	6.3 ; HLA-A3001	0.04 ; HLA-A3001	*n.a.*	1	3	4	69.208	0.79	4.65	no	not
IN_28_T1	IN_28_D	9	DTAPSGESR	T_A	intronic	14:103563126	APOPT1 + AL139300.1	pFind + PROSIT	StrelkaRNA	619.72 ; HLA-C1203	2.68 ; HLA-A3001	19019.6 ; HLA-C1203	12 ; HLA-C1203	*n.a.*	0.22	95	9	0.010	0.55	1.91	no	partly
IN_28_T1	IN_28_E	8	E*P*LTTREI	T_C	Ser167Pro	10:5452493	NET1	pFind	StrelkaRNA	1504.26 ; HLA-B0702	1.72 ; HLA-B0702	8641.6 ; HLA-B0702	6.5 ; HLA-B0702	*n.a.*	0.04	222	161	0	0.66	16.29	no	not
IN_28_T1	IN_28_F	9	GARLSSGRL	T_G	intronic	19:10116798	EIF3G	pFind	StrelkaRNA	851.18 ; HLA-B0702	1.41 ; HLA-B0702	673.2 ; HLA-B0702	1.3 ; HLA-B0702	*n.a.*	0.30	40	10	0.360	0.51	10.13	no	not
IN_28_T1	IN_28_G	11	VGSGLGPGWVM	G_C	intronic	11:724830	EPS8L2	pFind	StrelkaRNA	716.99 ; HLA-C1203	2.54 ; HLA-C0401	34650.4 ; HLA-B0702	42 ; HLA-B4403	*n.a.*	0.29	217	7	0	0.80	51.42	no	not
IN_30_T1	IN_30_A	10	**QCKRSSSSYR	delAA	Lys33 del	11:19143046	ZDHHC13	pFind + PROSIT	StrelkaRNA	24929.82 ; HLA-C0602	18.34 ; HLA-A2601	34030.6 ; HLA-A2601	55 ; HLA-A2601	*n.a.*	0.21	66	47	0	0.75	11.59	no	not
IN_32_T1	IN_32_A	10	AP*K*SSSGFSL	C_G	Asn24Lys	6:35555969	AL033519.2	pFind + PROSIT	StrelkaRNA	20.15 ; HLA-B0702	0 ; HLA-B0702	31.5 ; HLA-B0702	0.15 ; HLA-B0702	*n.a.*	0.75	11	4	26.887	0.87	10.25	no	not
IN_32_T1	IN_32_B	8	GP*G*SIQKR	A_G	Arg1172Gly	17:46330791	LRRC37A	pFind + PROSIT	StrelkaRNA	7184.09 ; HLA-B5601	4.91 ; HLA-B5601	36293.1 ; HLA-B0702	60 ; HLA-B0702	*n.a.*	0.11	31	36	0	0.85	16.72	no	partly
IN_32_T1	IN_32_C	10	ST*M*SALPNSR	A_T	Lys13Met	17:32876833	AC084809.1	pFind + PROSIT	StrelkaRNA	24.2 ; HLA-A1101	0.08 ; HLA-A1101	24.8 ; HLA-A1101	0.15 ; HLA-A1101	*n.a.*	0.13	520	38	0	0.42	2.41	no	not
IN_33_T1	IN_33_A	11	EA*E*VEESLGLR	A_G	Lys15Glu	3:194518211	LINC00884	pFind	StrelkaRNA	834.75 ; HLA-C1202	1.26 ; HLA-A2601	24262.4 ; HLA-A2601	17 ; HLA-A2601	*n.a.*	0.29	1	7	0.010	*n.a.*	*n.a.*	yes	not
IN_34_T1	IN_34_A	9	**SEVQDRAVP	insA	Arg188 ins	2:43225241	ZFP36L2	pFind + PROSIT	Mutect2	75.15 ; HLA-B4002	0.91 ; HLA-B4002	7869.7 ; HLA-B4002	6.5 ; HLA-B4002	0.16	*n.a.*	347	*n.a.*	0	0.89	2.12	no	fully
IN_36_T1	IN_36_A	8	AGLGGVKL	G_A	intronic	7:127588566	ARF5	pFind	StrelkaRNA	5242.68 ; HLA-B1402	4.45 ; HLA-B1402	35647.1 ; HLA-C0401	31 ; HLA-B3503	*n.a.*	0.75	90	4	0.857	0.57	52.62	*n.a.*	not
IN_37_T1	IN_37_A	8	A*T*ERKEAK	G_A	Ala194Thr	X:53413267	SMC1A	pFind	StrelkaRNA	14230.63 ; HLA-C1504	9.59 ; HLA-A0301	20666.9 ; HLA-A0301	19 ; HLA-A0301	*n.a.*	0.40	108	5	0	0.72	10.06	no	partly
IN_37_T1	IN_37_B	9	DVVVVH*R*RR	G_A	Gly43Arg	3:38136909	ACAA1	pFind + PROSIT	StrelkaRNA+Mutect2	28.71 ; HLA-A6801	0.21 ; HLA-A6801	46.7 ; HLA-A6801	0.6 ; HLA-A6801	0.13	0.23	39	39	0	0.94	66.82	yes	partly
IN_37_T1	IN_37_C	8	G*S*PSLSQR	C_T	Pro208Ser	1:154959378	PYGO2	pFind	StrelkaRNA	6688.81 ; HLA-A6801	4.56 ; HLA-A6801	20198 ; HLA-A6801	17 ; HLA-A6801	*n.a.*	0.28	114	25	0	0.32	4.62	no	not
IN_37_T1	IN_37_D	8	KFAQK*V*LR	A_G	Met41Val	1:96679500	RPL7P9	PROSIT	StrelkaRNA	21753.84 ; HLA-A6801	8.6 ; HLA-A0301	17518.2 ; HLA-A0301	15 ; HLA-A0301	*n.a.*	0.003	139	1389	0.010	0.61	14.91	no	not
IN_37_T1	IN_37_E	11	RLANTQ*A*KKAK	G_C	Gly164Ala	6:44396392	CDC5L	pFind	StrelkaRNA	42.35 ; HLA-A0301	0.25 ; HLA-A0301	143.7 ; HLA-A0301	0.6 ; HLA-A0301	*n.a.*	0.12	137	24	0	0.72	0.20	no	fully
IN_37_T1	IN_37_F	10	SAADVVVVH*R*	A_G	Gly43Arg	3:38136909	ACAA1	pFind + PROSIT	StrelkaRNA+Mutect2	17.27 ; HLA-A6801	0.01 ; HLA-A6801	13.9 ; HLA-A6801	0.18 ; HLA-A6801	0.13	0.23	39	39	0	0.93	2.12	no	fully
IN_37_T1	IN_37_G	13	TVG*V*PTVLEKLQK	T_G	Leu672Val	3:185192384	EHHADH	pFind	StrelkaRNA	18184.72 ; HLA-A0301	6.33 ; HLA-A0301	1618.5 ; HLA-A0301	3 ; HLA-A0301	*n.a.*	0.14	206	22	0	0.16	63.81	no	not
IN_37_T1	IN_37_H	8	VDAN*R*KIY	G_A	Gly107Arg	11:18514716	TSG101	PROSIT	StrelkaRNA	4514.79 ; HLA-B1801	1.39 ; HLA-B1801	20875.5 ; HLA-B1801	14 ; HLA-B1801	*n.a.*	0.03	65	213	0	0.58	5.41	no	partly
IN_38_T1	IN_38_A	9	DVI*R*KALQY	G_A	Gly926Arg	1:97082461	DPYD	pFind + PROSIT	StrelkaRNA+Mutect2	19.45 ; HLA-A2601	0 ; HLA-A2601	25.4 ; HLA-A2601	0.04 ; HLA-A2601	0.44	0.28	238	596	0	0.78	0.56	no	fully
IN_38_T1	IN_38_B	8	RP*H*VGIHL	T_C	Tyr243His	20:32228420	POFUT1	pFind + PROSIT	StrelkaRNA+Mutect2	29.72 ; HLA-B0702	0.09 ; HLA-B0702	139.2 ; HLA-B0702	0.6 ; HLA-B0702	0.40	0.44	164	587	0	0.94	3.25	no	fully
IN_38_T1	IN_38_C	8	SITPGT*V*L	A_G	Ile149Val	12:112406782 + 4:65573906	RPL6 + AC115223.1	pFind	StrelkaRNA+Mutect2	46.21 ; HLA-C0102	0.34 ; HLA-C0102	7714 ; HLA-B0702	6 ; HLA-B0702	0.16	0.07	67	17892	0	0.18	1.31	no	not
IN_38_T1	IN_38_D	11	SQSTTASL*F*KK	C_T	Ser451Phe	1:151034224	PRUNE1	pFind + PROSIT	StrelkaRNA+Mutect2	42.78 ; HLA-A1101	0.32 ; HLA-A1101	68.8 ; HLA-A1101	0.5 ; HLA-A1101	0.30	0.37	43	142	0	0.85	1.57	no	fully
IN_38_T1	IN_38_E	9	STTASL*F*KK	C_T	Ser451Phe	1:151034224	PRUNE1	pFind + PROSIT	StrelkaRNA+Mutect2	16.54 ; HLA-A1101	0 ; HLA-A1101	8.2 ; HLA-A1101	0.02 ; HLA-A1101	0.30	0.37	43	142	0	0.75	2.48	no	fully

6.4 Immunogenicity data summary

Immunogenicity assessment data from all reactive neoantigen candidates. Modified acDC assays were performed using patient derived PBMC or TILs (see Figure 25 b) and analysed by ELIspot. IFN-g spot forming units (SFU) for T cells tested against the mutated peptide (test condition) and an irrelevant peptide (control condition) are reported. The mean SFU over all replicates (duplicates or triplicates, where available) was calculated per condition and the ratio as well as the difference (delta) of mean SFU was determined. Shown are neoantigen candidates with a ratio of SFU > 2 and the delta of SFU > 50, defined as immunogenic. The target cell used for pulsing (1 µM synthetic peptide) and presentation or the neoantigen or irrelevant peptide is shown. The respective irrelevant peptide utilized is also annotated. T cells non-specifically stimulated with 0.5 ng/µl PMA and 1 ng/µl Ionomycin were used as positive control, and only TCM was added to the T cells as negative control. LCL, lymphoblastoid cell line; PBMCs, peripheral blood mononuclear cells; SFU, spot forming units; TIL, tumor-infiltration lymphocytes; wt, wild type.

	Peptide ID	IN_01_A	IN_01_B	IN_01_C	IN_04_A	IN_04_I	IN_04_J	IN_04_J	IN_04_K	IN_04_M	IN_05_A	IN_11_B	IN_11_D
	assay	new_10	new_1	new_10	new_5	new_5	new_5	new_5	new_5	new_5	new_4	new_3	new_3
	target cells	LCL IN-01	LCL IN-01	LCL IN-01	LCL IN-04	LCL IN-04	LCL IN-04	LCL IN-04	LCL IN-04	LCL IN-04	LCL IN-11	LCL IN-11	LCL IN-11
	irrelevant peptide	IN_01_wt	IN_01_wt	IN_01_wt	IN_4_wt	IN_4_wt	IN_4_wt	IN_4_wt	IN_4_wt	IN_4_wt	IN_11_wt	IN_11_wt	IN_11_wt
	sample type	TIL	PBMC	TIL	PBMC	PBMC	PBMC	PBMC	PBMC	PBMC	PBMC	PBMC	PBMC
	PBMC/TIL aliquot	IN-1_OP1	IN-01.1	IN-1_OP1	IN-04.2	IN-04.2	IN-04.2	IN-04.2	IN-04.2	IN-04.2	IN-05.1	IN-11.3	IN-11.3
	method day 1	non-enriched	enriched	enriched	enriched	enriched	enriched	non-enriched	non-enriched	non-enriched	enriched	enriched	enriched
Number of SFU	mutated	19	78	138	156	108	269	24	602	97	101	824	121
	mutated	176	64	232	193	50	606	148	613	161	164	466	406
	mutated	34	n.a.	306	200	57	608	349	509	88	n.a.	n.a.	n.a.
	irrelevant	22	10	191	72	27	399	50	56	42	23	243	139
	irrelevant	13	8	84	34	13	30	1	259	20	100	82	81
	irrelevant	11	n.a.	52	69	18	31	13	511	48	n.a.	n.a.	n.a.
	positive	1500	612	1500	1800	1800	1800	794	1800	1800	939	932	1234
	negative	n.a.	0	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	1	0	1
	mean mutated SFU	76	71	225	183	72	494	174	575	115	133	645	264
	mean irrelevant SFU	15	9	109	58	19	153	21	275	37	62	163	110
	Delta mutated-irrelevant SFU	**61**	**62**	**116**	**125**	**52**	**341**	**152**	**299**	**79**	**71**	**483**	**154**
	Ratio mutated/irrelevant SFU	**4.98**	**7.89**	**2.07**	**3.14**	**3.71**	**3.22**	**8.14**	**2.09**	**3.15**	**2.15**	**3.97**	**2.40**

	Peptide ID	IN_19_B	IN_19_C	IN_19_E	IN_19_F	IN_19_H	IN_19_I	IN_19_I	IN_22_A	IN_33_A	IN_37_B
	assay	new_1	new_1	new_1	new_1	new_1	new_1	new_1	new_7	new_4	new_6
	target cells	LCL IN-19	LCL IN-19	LCL IN-19	LCL IN-19	LCL IN-19	LCL IN-19	LCL IN-19	LCL IN-22	LCL HD07	LCL IN-37
	irrelevant peptide	IN_01_wt	IN_01_wt	IN_01_wt	IN_01_wt	IN_01_wt	IN_01_wt	IN_01_wt	IN_38_wt	IN_01_wt	IN_38_wt
	sample type	PBMC	PBMC	PBMC	PBMC	PBMC	PBMC	PBMC	TIL	PBMC	PBMC
	PBMC/TIL aliquot	IN-19.1	IN-19.1	IN-19.1	IN-19.1	IN-19.1	IN-19.1	IN-19.1	IN-22_OP1	IN-33.1	IN-37.1
	method day 1	non-enriched	non-enriched	enriched	non-enriched	enriched	enriched	non-enriched	enriched	enriched	enriched
Number of SFU	mutated	33	629	333	683	520	247	606	137	29	54
	mutated	501	716	300	711	630	137	413	278	127	714
	mutated	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	297	n.a.	378
	irrelevant	151	30	130	184	359	69	149	103	26	136
	irrelevant	57	347	177	251	184	41	133	174	28	314
	irrelevant	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	76	n.a.	21
	positive	964	1447	1111	1569	1500	1586	1740	1500	1000	550
	negative	1	0	0	0	0	0	0	n.a.	1	n.a.
	mean mutated SFU	267	673	317	697	575	192	510	237	78	382
	mean irrelevant SFU	104	189	154	218	272	55	141	118	27	157
	Delta mutated-irrelevant SFU	**163**	**484**	**163**	**480**	**304**	**137**	**369**	**120**	**51**	**225**
	Ratio mutated/irrelevant SFU	**2.57**	**3.57**	**2.06**	**3.20**	**2.12**	**3.49**	**3.61**	**2.02**	**2.89**	**2.43**

6.5 List of figures

6.6 List of tables

6.7 Attributions

The majority of the presented data in this thesis was published by Tretter et al. (2023) and permission for usage was kindly granted by Springer Nature. Parts of the presented data within this thesis was generated in cooperation with many research groups and institutions and the contribution of each person and/or group will be listed here.

Clinical data:
Follow-up clinical data was collected and summarized by Clara von Frankenberg as shown in Figure 6 and Figure 7 and Appendix 6.1 and 6.2.

Phenotyping data:
Anja Stelzl, Eva Bräunlein, Sabine Mall and Stefanie Stein (AG Krackhardt, IIIrd Medical Department, Klinikum rechts der Isar, School of Medicine, Technical University of Munich, Munich, Germany) helped with tumor sample and PBMC sample collection, processing and flow-cytometry measurements for some experiments shown in Figure 9, Figure 10 and Figure 11.
RNA sequencing library preparation and sequencing of sorted TILs in Figure 13 was performed by AG Klink (Institute for Clinical Genetics, Technical University Dresden, Germany). Raw data processing and data analysis was performed by Thomas Engleitner (AG Rad, Center for Translational Cancer Research, School of Medicine, Technical University of Munich, Munich, Germany and Institute of Molecular Oncology and Functional Genomics, School of Medicine, Technical University of Munich, Munich, Germany) and Niklas de Andrade Krätzig (AG Rad, IInd Medical Department, Klinikum rechts der Isar, School of Medicine, Technical University of Munich, Munich, Germany; Center for Translational Cancer Research, School of Medicine, Technical University of Munich, Munich, Germany and Institute of Molecular Oncology and Functional Genomics, School of Medicine, Technical University of Munich, Munich, Germany).

Genomic and transcriptomic data:
Library preparation, DNA sequencing and RNA sequencing of tumor samples and PBMCs was performed by the DKTK and DKFZ, Heidelberg, Germany. Mutation calling was performed by Sebastian Lange (AG Rad, IInd Medical Department, Klinikum rechts der Isar, School of Medicine, Technical University of Munich, Munich, Germany; Center for Translational Cancer Research, School of Medicine, Technical University of Munich, Munich, Germany andInstitute of Molecular Oncology and Functional Genomics, School of Medicine, Technical University of Munich, Munich, Germany) as shown in Figure 15, Figure 16 and Figure 17.

Immunopeptidomics data set:
Immunoprecipitation and MS measurement of HLA class I bound peptides was performed by Matteo Pecoraro (AG Mann, Department of Proteomics and Signal Transduction, Max Plank Institute of Biochemistry, Munich, Germany) as shown in Figure 18, Figure 19, Figure 20, Figure 21 and Figure 22.

Neoantigen candidate analysis:
The patient variant reference data set for neoantigen identification in Figure 23 was generated by Niklas de Andrade Krätzig.
Prosit analysis for neoantigen identification was performed by Mathias Wilhelm (AG Küster, Chair of Proteomics and Bioanalytics, School of Life Sciences, Technical University of Munich, Freising, Germany and Computational Mass Spectrometry, School of Life Sciences, Technical University of

Munich, Freising, Germany) and Daniel Zolg (AG Küster, Chair of Proteomics and Bioanalytics, School of Life Sciences, Technical University of Munich, Freising, Germany).
The prediction pipeline used in Figure 24 was developed together with Niklas de Andrade Krätzig.

Immunogenicity assessment data:
Three acDC experiments shown in Figure 25 b were performed in cooperation with Philipp Siefert (AG Krackhardt, IIIrd Medical Department, Klinikum rechts der Isar, School of Medicine, Technical University of Munich, Munich, Germany). MHCFlurry and NetHMC predictions were performed by Philipp Seifert (see Appendix 6.3).
Healthy donor immunogenicity tests were performed by Johannes Untch (AG Krackhardt, IIIrd Medical Department, Klinikum rechts der Isar, School of Medicine, Technical University of Munich, Munich, Germany) as shown in Figure 25 c.

Peptide verification:
Spectra prediction used for Figure 28 and Figure 32 as well as retention time prediction in Figure 29 using Prosit was performed by Mathias Wilhelm and Daniel Zolg.
Synthetic peptide measurements used for Figure 28 and Figure 32 were performed by Matteo Pecoraro, Mathias Wilhelm and Daniel Zolg.
Spectral angle calculation of predicted, synthetic and experimental peptides used in Figure 28 and Figure 32 were performed by Mathias Wilhelm and Daniel Zolg.
GTEx data analysis in Figure 30, Figure 31 and Figure 32 was performed by Niklas de Andrade Krätzig.

General:
Michael Hiltensperger reformatted most graphs using Illustrator and generated Figure 8 and Figure 32. Philipp Seifert developed parts of the analysis scripts shown in Appendix 6.9.1, 6.9.2 and 6.9.4.

6.8 Acknowledgements

First, I would like to express my sincere gratitude to my supervisor Prof. Angela Krackhardt for giving me the opportunity to perform this compelling research project. Her excellent supervision and guidance have been invaluable throughout the whole process of this work. It has been a pleasure for me to work in her group and I am very grateful for all the helpful scientific input and feedback.

Furthermore, I would like to thank the members of my thesis advisory committee, Prof. Julien Gagneur and Prof. Wilko Weichert, for their insightful discussions and their constructive input on the project. Special thanks go to Prof. Gagneur for introducing me to Prof. Küster and his work on Prosit, which substantially influenced the quality of my thesis. Also, I want to thank Prof. Weichert, who sadly prematurely left us, for his feedback as my mentor and for everything the Department of Pathology contributed to my work.

I would like to express my deepest gratitude to all the patients and their families for being part of this study and all the DKTK partner site clinics for collecting and shipping the samples. Furthermore, this project would not have been possible without the DKFZ and the DKTK who generously funded this research and all the great collaboration partners across DKTK who supported this work.

A very special thanks goes to my co-first author Niklas de Andrade Krätzig for all his help with my R scripts and analysis, for all the great and inspiring discussions and all the contributions he made towards the success of this project. I would also like thank the whole bioinformatics team of Prof. Roland Rad, Thomas Engleitner and Sebastian Lang, for their great work and support.

I am also very grateful to Michael Hiltensperger for his amazing scientific and personal guidance throughout the publication process of our paper, for all the great ideas, constructive feedback, amazing Illustrator figures and generally all his efforts and time.

Special thanks go to all the AG Krackhardt lab members, Anja Stelzl, Dario Gosmann, Gaia Lupoli, Philipp Seifert, Johannes Untch, Clara von Frankenberg, Stefanie Stein, Franziska Füchsl, Cigdem Atay and Eva Bräunlein, for creating a great atmosphere at work and for their scientific and emotional support during our time together. Especially, I would like to thank Anja Stelzl for teaching me a lot when I started, for her great help in sample processing, including long nights in the lab, and her moral support. Also, I want to thank Dario Gosmann, Gaia Lupoli and Johannes Untch for their emotional and scientific support and all the fun we had together. I also especially thank Philipp Seifert for his great help in implementing, improving and running acDC assays with me.

Lastly, I want to express my heartfelt thanks to my husband, Timo Tretter, my family, and all friends for their unwavering support, patience, and optimism throughout my time in the lab and the many weekends of writing. Thank you for being my personal anchors.

6.9 R studio scripts

6.9.1 Analysis sequencing data

```
library(tidyverse)
library(openxlsx)
library(stringr)
library(reshape2)
library(ggplot2)
library(extrafont)
library(ggrepel)
library(data.table)
library(RColorBrewer)
library(dplyr)
library(scales)
library(writexl)
library(ggpubr)
library(ComplexUpset)
library(rlist)

#### (1) #### import all .tsv files from folder and add to one dataframe ________#######
path.tsv.files="mutation_calling/raw_data/2020_05_08/"
tsv.files=list.files(path=path.tsv.files, pattern = "*.tsv", full.names = T)
all.patients_DF_new = plyr::ldply(tsv.files, read.delim)# fread

#### (2) #### correct TumorVF & NormalVF _________________________________######
all.patients_DF_new <- mutate(all.patients_DF_new, TumorVF=TumorAD/(TumorAD+TumorRD))
all.patients_DF_new <- mutate(all.patients_DF_new, NormalVF=NormalAD/(NormalAD+NormalRD))

#### (3) #### Import references ___________________________________________######
source(file = "functions/import.references.R")

all.patients_DF_new$patientID <- sub("Mel15OP1","Mel15_T1", all.patients_DF_new$patientID)
all.patients_DF_new$patientID <- sub("Mel15OP2","Mel15_T2", all.patients_DF_new$patientID)
all.patients_DF_new$patientID <- sub("Q1PB42_T1","Q1PB42_T3", all.patients_DF_new$patientID)

all.patients_DF_new <- all.patients_DF_new %>% rename(Master_ID=patientID)
all.patients_DF_new$Master_ID_group <- as.factor(str_sub(all.patients_DF_new$Master_ID,1,6))
all.patients_DF_new$Master_ID_group <- sub("_","", all.patients_DF_new$Master_ID_group)
## merge with references !!! IN-25_T2 is lost because doesn't exist, IN-39 and IN-40 are lost because not part of final cohort because no MS data!!!
all.patients_DF_new <- merge(reference.master, all.patients_DF_new, by.x = "Master_ID", by.y = "Master_ID_group") %>%
  select(-Master_ID) %>%
  rename(Master_ID = Master_ID.y) %>%
  merge(reference.entity) %>%
  mutate(Tumor_entity=str_replace_all(Tumor_entity, c("nonseminomatous germ cell tumor"="Non-sem. germ cell tumor",
                                   "Desmoplastic small-round-cell tumor"="Desmopl.small-round-cell tumor",
                                   "atypical carcinoid of the lung"="Atypical lung carcinoid",
                                   "Mukoedidermoid Carcinoma"="Mukoepidermoid Carcinoma",
                                   "Urothelcacrinoma"="Urothelcarcinoma",
                                   "adrenocortical carcinoma"="Adrenocortical carcinoma"))) %>%
  mutate(EFFECT=str_replace_all(EFFECT, c("non_coding_transcript_exon_variant"="Non-coding transcript exon variant",
                             "missense_variant"="Missense variant",
                             "splice_donor_variant"="Splice donor variant",
                             "splice_acceptor_variant"="Splice acceptor variant",
                             "non_coding_transcript_exon_variant"="Non-coding transcript exon variant",
                             "stop_gained"="Stop gained",
                             "splice_donor_variant&intron_variant"="Splice donor variant & intron variant",
                             "frameshift_variant"="Frameshift variant",
                             "disruptive_inframe_deletion"="Disruptive inframe deletion",
                             "splice_acceptor_variant&intron_variant"="Splice acceptor variant & intron variant",
                             "Splice donor variant&intron_variant"="Splice donor variant & intron variant",
                             "Splice acceptor variant&intron_variant"="Splice acceptor variant & intron variant"))) %>%
  mutate(geneBiotype=str_replace_all(geneBiotype, c("3prime_overlapping_ncRNA"="3'-overlapping ncRNA",
                                     "antisense" = "Antisense",
                                     "processed_pseudogene"="Processed Pseudogene",
                                     "protein_coding"="Protein Coding",
                                     "transcribed_Processed Pseudogene"="Processed Pseudogene (transcribed)",
                                     "unProcessed Pseudogene"="Unprocessed Pseudogene",
                                     "sense_intronic"="Sense Intronic",
                                     "transcribed_Unprocessed Pseudogene;processed_transcript"="Unprocessed Pseudogene (transcribed) + pt",
                                     "transcribed_Unprocessed Pseudogene"="Unprocessed Pseudogene (transcribed)",
                                     "unitary_pseudogene"="Unitary Pseudogene",
                                     "processed_transcript"="Processed Transcript",
                                     "IG_V_pseudogene"="Variable chain IG Pseudogene",
                                     "sense_overlapping"="Sense overlapping")))
```

```
all.patients_DF_new["Metastasis"] <- str_sub(all.patients_DF_new$Master_ID, 8,9)
all.patients_DF_new$Metastasis <- sub("^1","T1", all.patients_DF_new$Metastasis)
all.patients_DF_new$Metastasis <- sub("^2","T2", all.patients_DF_new$Metastasis)
all.patients_DF_new$Tumor_ID <- paste(all.patients_DF_new$Patient_ID, all.patients_DF_new$Metastasis, sep = "_")

#### (4) #### general modifications of data set and grouping _______________________________________________#####
## create new columns for Mutation_ID and Biotype_group
all.patients_DF_new$Mutation_ID <- paste(all.patients_DF_new$CHROM, all.patients_DF_new$POS,all.patients_DF_new$REF,
all.patients_DF_new$ALT, sep = "_")
all.patients_DF_new$Biotype_group <- paste(all.patients_DF_new$geneBiotype)
all.patients_DF_new$Biotype_group <- as.character(all.patients_DF_new$Biotype_group)

fwrite(all.patients_DF_new[!all.patients_DF_new$Patient_ID == "Mel15",], file =
"mutation_calling/Results_export/Mutations_all_table_allINPatients_new_20220310.csv")
#all.patients_DF_new <- read.csv(file = "mutation_calling/Results_export/Summary_table_allPatients_new_20210917.csv")

## change DF into DT for better handling and combine Biotypes into major groups
all.patients_DT_new <- data.table(all.patients_DF_new)
all.patients_DT_new[grep("pseudogene", Biotype_group, ignore.case = T), Biotype_group := "Pseudogene"] # combine all pseudogene variants
all.patients_DT_new[Biotype_group %in% c("bidirectional_promoter_lncRNA", "macro_lncRNA"), Biotype_group := "lncRNA"] #combine lncRNAs
all.patients_DT_new[Biotype_group %in% c("Sense overlapping", "IG_V_gene", "IG_C_gene", "TR_V_gene", "non_coding"), Biotype_group :=
"Others"] # combine various smaller subtypes
all.patients_DT_new[Biotype_group %in% c("3'-overlapping ncRNA", "misc_RNA", "lncRNA", "snoRNA", "snRNA", "miRNA", "scaRNA", "rRNA",
"vaultRNA", "lincRNA", "Antisense"), Biotype_group := "Regulatory RNAs"] #combine all RNA subtypes

all.patients_DT_new <- all.patients_DT_new[!Master_ID %in% c( "Mel15_T1", "Mel15_T2" )]
all.patients_DF_new <- data.frame(all.patients_DT_new)

# !!!!!! wrong naming with unique Pep --> should be unique Mut
## collapse data table with unique Mutation_ID for each Tumor_ID
othercols <- c("CHROM", "POS", "REF", "ALT", "Tumor_ID")
mergecols <- setdiff(names(all.patients_DT_new), othercols)
all.patients_DT_uniqueMut <- all.patients_DT_new[, lapply(.SD, function(x){paste0(unique(x),collapse=";")}), .SDcols = mergecols, by=othercols]

fwrite(all.patients_DT_uniqueMut, file = "mutation_calling/Results_export/Summary_table_allPatients_uniqueMut_new_20210930.csv")

# DEBUG check if unique entries/number if mutations per Tumor are the same
all.patients_DT_uniqueMut[Tumor_ID == "IN_01_T1"]
tmp <- all.patients_DT_uniqueMut[Tumor_ID == "IN_01_T1"]
setorder(tmp, CHROM, POS)
tmp
all.patients_DT[POS == 90789]

mutations_counts <- all.patients_DT_uniqueMut[, .N, by=.(Tumor_ID)]
mutations_counts_2 <- all.patients_DT_uniqueMut[, .(number_of_distinct_mutations = uniqueN(Mutation_ID)), by = Tumor_ID]
mutations_counts_3 <- all.patients_DT_new[, .(number_of_distinct_mutations = uniqueN(Mutation_ID)), by = Tumor_ID]

# DEBUG adapt Biotype_group again after collapsing
all.patients_DT_uniqueMut[grep("Protein Coding", geneBiotype, ignore.case = T), Biotype_group := "Protein Coding"]
all.patients_DT_uniqueMut[, Biotype_group := str_replace(Biotype_group, ";TEC", "")]
all.patients_DT_uniqueMut[, Biotype_group := str_replace(Biotype_group, "TEC;", "")]
all.patients_DT_uniqueMut[, Biotype_group := str_replace(Biotype_group, "Others;", "")]
all.patients_DT_uniqueMut[, Biotype_group := str_replace(Biotype_group, ";Others", "")]
all.patients_DT_uniqueMut[grep("Pseudogene;Processed Transcript", Biotype_group, ignore.case = T), Biotype_group := "Processed
Transcript;Pseudogene"]
all.patients_DT_uniqueMut[grep("Pseudogene;Regulatory RNAs", Biotype_group, ignore.case = T), Biotype_group := "Regulatory
RNAs;Pseudogene"]
all.patients_DT_uniqueMut[grep("Sense Intronic;Regulatory RNAs", Biotype_group, ignore.case = T), Biotype_group := "Regulatory RNAs;Sense
Intronic"]
all.patients_DT_uniqueMut[grep("Processed Transcript;Regulatory RNAs", Biotype_group, ignore.case = T), Biotype_group := "Regulatory
RNAs;Processed Transcript"]

all.patients_DT_uniqueMut[SOURCE %in% c("Mutect2;StrelkaRNA", "StrelkaRNA;Mutect2"), SOURCE := "Mutect2+StrelkaRNA"]
all.patients_DT_uniqueMut[grep("Mutect2", SOURCE, ignore.case = T), Mutation_group := "Somatic"]
all.patients_DT_uniqueMut[grep("Mutect2+", SOURCE, ignore.case = T), Mutation_group := "Somatic"]
all.patients_DT_uniqueMut[grep("^StrelkaRNA", SOURCE, ignore.case = T), Mutation_group := "RNA editing"]

#exclude Mel15 samples
all.patients_DT_uniqueMut <- all.patients_DT_uniqueMut[!Master_ID %in% c( "Mel15_T1", "Mel15_T2" )]
all.patients_DF_uniqueMut <- data.frame(all.patients_DT_uniqueMut)
#select core samples
all.patients_DT_uniqueMut_Core <- all.patients_DT_uniqueMut[!Master_ID %in% c("GXL1B7_T2", "64EMZ9_T1", "Q1PB42_T1", "Q1PB42_T3",
"1MULDR_T1","1MULDR_T2","1MULDR_T3","NVDER5_T1", "LFNUX6_T2", "ATE46U_T1", "Mel15_T1", "Mel15_T2" )]
all.patients_DF_uniqueMut_Core <- data.frame(all.patients_DT_uniqueMut_Core)
```

```
## subset dataset to subgroups: e.g. somatic and RNA Mutations
all.patients_DF_DNA <- subset(all.patients_DF_uniqueMut, Mutation_group == "Somatic")
all.patients_DF_RNA <- subset(all.patients_DF_uniqueMut, Mutation_group == "RNA editing")
all.patients_DF_RNA <- subset(all.patients_DF_uniqueMut, SOURCE %in% c("StrelkaRNA","Mutect2+StrelkaRNA"))
all.patients_DF_uniqueMut_DNAcoverage <-
as.data.table(fread("mutation_calling/Results_export/Mutations_all_table_allINPatients_new_20220310_coverage.tsv"))

# (5) #### Processing Data and plots ____________________________________________________________####
### 1 ### Top mutation containing genes ::geneDF:: ________________________________________________####
## filter data set if needed
all.patients_DF_IN <- all.patients_DF_new[!all.patients_DF_new$Patient_ID == "Mel15",]
all.patients_DF_IN_DNA <- subset(all.patients_DF_IN, SOURCE == "Mutect2")

## ::geneDF1:: ### 25 Genes, with the most mutations (unique mutations; meaning: same mutation in different patients will be counted as 1)
geneDF1 <- all.patients_DF_IN %>%
  distinct(CHROM,POS,REF,ALT,GENE) %>%
  group_by(GENE) %>%
  summarise(N.total=n()) #%>%
  #top_n(15, N.total)
## ::geneDF2:: ### as geneDF, but with implemented condition SOURCE for distinction of origin of mutation
geneDF2 <- all.patients_DF_IN %>%
  distinct(CHROM,POS,REF,ALT,GENE,SOURCE) %>%
  group_by(GENE,SOURCE) %>%
  summarise(N=n()) %>%
  spread(key = SOURCE, value = N, fill = 0) %>%    # from long to wide
  rename(N.Mutect2=Mutect2, N.Strelka=StrelkaRNA)
## ::geneDF3:: ### as geneDF2, but with information for mutations found by both tools
geneDF3 <- all.patients_DF_IN %>%
  distinct(CHROM,POS,REF,ALT,GENE,SOURCE) %>%
  group_by(CHROM,POS,REF,ALT,GENE) %>%
  summarise(SOURCE=SOURCE %>% unique %>% sort %>% paste(collapse = " + ")) %>%
  ungroup() %>%
  group_by(GENE,SOURCE) %>%
  summarise(N=n())
## ::geneDF:: ### both information together
geneDF <- merge(geneDF2, geneDF1) %>%
  top_n(25, N.total) %>%
  #filter(N>24) %>%
  arrange(desc(N.total))

geneDF.overlap <- merge(geneDF3, geneDF1) %>%
  top_n(38, N.total) %>%
  arrange(desc(N.total))

# transform to long format for plotting
geneDF.long <- geneDF %>%
  rename(Mutect2=N.Mutect2, Strelka=N.Strelka, total=N.total) %>%
  gather(key = Source, N, Mutect2:total) %>%
  filter(Source!="total") # exclude rows that contain values for N.total

######## 1a ___ Plots ______________________________________________________________________#####
ggplot(geneDF, mapping = aes(x=reorder(GENE,N.total), y=N.total))+
  geom_col()+
  coord_flip()+
  labs(y="Total number of different mutations", x="Gene")+
  theme_PS()

ggplot(geneDF.long, mapping = aes(x=reorder(GENE,N), y=N, fill=Source))+
  geom_bar(position = "dodge", stat="identity")+
  coord_flip()+
  labs(y="Total number of different mutations", x="Gene", fill="Tool for \nmutation calling")+
  theme_PS()+
  theme(legend.position = c(.8,.6))

Top20_biotypes <- read.csv("mutation_calling/Results_export/Top20_mutGenes_Biotypes.csv", sep = ";")
geneDF.overlap <- merge(geneDF.overlap, Top20_biotypes, by= "GENE")
ggplot(geneDF.overlap, mapping = aes(x=reorder(GENE,N.total), y=N, fill=SOURCE))+
  geom_col(position = "stack")+
  coord_flip()+
  labs(y="Number of unique mutations", x="Gene", fill="Mutation origin")+
  theme_PS()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size= 25),
        axis.text.y = element_text(size= 15),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size= 25),
        legend.title = element_text(size=25, face = "bold"))+
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA"))

### other analysis and plot for Top mutated genes
Top_mut_genes <- as.vector(unique(geneDF.overlap$GENE))
all.patients_DF_uniquePep_topMutGenes <- subset(all.patients_DF_uniquePep, GENE %in% Top_mut_genes)

### filled by genes
nb.cols <- 20
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.cols)
ggplot(all.patients_DF_uniquePep_topMutGenes[!all.patients_DF_uniquePep_topMutGenes$Patient_ID == "Mel15",],
aes(x=forcats::fct_infreq(Tumor_ID), fill=GENE))+
  geom_bar()+
  coord_flip()+
  labs(y="Number of unique mutations", x="Patient", fill="Gene")+
  theme_PS()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size= 25),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size= 25),
        legend.title = element_text(size=25, face = "bold"))+
  scale_fill_manual(values = mycolors)

### filled by Patients
nb.cols <- 27
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.cols)
ggplot(all.patients_DF_uniquePep_topMutGenes[!all.patients_DF_uniquePep_topMutGenes$Patient_ID == "Mel15",],
aes(x=forcats::fct_infreq(GENE), fill=Patient_ID))+
  geom_bar(colour = "black")+
  coord_flip()+
  labs(y="Number of unique mutations per Patient", x="Gene", fill="Patient ID")+
  theme_PS()+
  scale_y_continuous(breaks = c(100, 200, 300, 400, 500))+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size= 25),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size= 25),
        legend.title = element_text(size=25, face = "bold"))+
  guides(fill=guide_legend(ncol=7))+
  scale_fill_manual(values = mycolors)

### 2 ### ::QA:: ____________________________________________________________________________#####
# Quality assessment of Peptides with Tumor_VF
QA <- distinct(all.patients_DF,CHROM, POS, REF, ALT, Master_ID, .keep_all = T) %>%
  select(Master_ID,
CHROM,POS,REF,ALT,GENE,AF,TumorVF,TumorAD,TumorRD,NormalAD,NormalRD,EFFECT,seqType,SOURCE,mutationType,mutationSubType) %>%
  filter((TumorAD+TumorRD)>19)

ggplot(QA)+
  geom_freqpoly(mapping=aes(TumorAD), binwidth = 500)+
  coord_cartesian(xlim=c(0,10000))+
  scale_y_log10()+
  xlim(0,11000)+
  labs(title="Mutation data validity", x="Number of Reads", y="density distribution")+
  theme_PS()

### Quality assessment of mutations according to TumorVF etc.
ggplot(QA, mapping = aes(TumorVF))+
  #geom_freqpoly(binwidth=0.01, color='red')+
  geom_histogram(binwidth=0.005)+
  coord_cartesian(ylim = c(0, 800))+
  theme_PS()

###__ 2a ### Filtering DNA Mutations__________________________________________________________________#####
## all DNA mutations
all.patients_DF_DNA <- as.data.table(subset(all.patients_DF_new, SOURCE == "Mutect2"))
## DNA-only mutations
all.patients_DF_DNA <- as.data.table(subset(all.patients_DF_uniqueMut, SOURCE == "Mutect2"))
```

```
col_names <- colnames(all.patients_DF_DNA)
changeCols_2 <- col_names[! col_names %in%
c("CHROM","POS","REF","ALT","Tumor_ID","Master_ID","Patient_ID","GENE","EFFECT","FEATUREID","HGVS_C","SOURCE","geneBiotype","transcrip
tBiotype",
"mutationType","Tumor_entity","Tumor_entity_short","Tumor_state","Metastatic_site","Tumor_origin","Metastasis","Mutation_ID","Biotype_grou
p","Mutation_group")]
all.patients_DF_DNA <- all.patients_DF_DNA[,(changeCols_2):= lapply(.SD, as.numeric), .SDcols = changeCols_2] # change to numeric values
all.patients_DF_DNA <- mutate(all.patients_DF_DNA, Coverage=TumorAD+TumorRD)
all.patients_DF_DNA_filtered <- as.data.table(all.patients_DF_DNA)
#different ways to filter --> select one
all.patients_DF_DNA_filtered <- all.patients_DF_DNA_filtered[TumorVF >=0.05 & Coverage >= 5 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_DNA_filtered <- all.patients_DF_DNA_filtered[Coverage >= 5 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_DNA_filtered <- all.patients_DF_DNA_filtered[TumorVF >=0.05 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_DNA_filtered <- all.patients_DF_DNA_filtered[TumorAD >=2 & NormalAD <= 1]
#mutations_counts_RNAediting_filtered <- all.patients_DF_RNA_small_filtered[, .N, by=.(Tumor_ID)]

## plot: compare un-filtered and filtered data sets
ggplot(all.patients_DF_DNA, aes(x = Tumor_ID)) +
  geom_bar(data = all.patients_DF_DNA, aes(x = Tumor_ID), stat = "count" , fill = "light grey") +
  geom_bar(data = all.patients_DF_DNA_filtered, aes(x = Tumor_ID), stat = "count")+
  #scale_y_continuous(limits = c(0,1.1*max(mutational_load.permaster_bothtools$N)), breaks =
seq(0,1.1*max(mutational_load.permaster_bothtools$N), by = 10000))+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")

all.patients_DF_DNA_filtered[, .N, by=.(Tumor_ID)]

###__2b ### Filtering RNA Mutations ______________________________________________________________________#####
## all RNA mutations
all.patients_DF_RNA <- as.data.table(subset(all.patients_DF_new, SOURCE == "StrelkaRNA"))

col_names <- colnames(all.patients_DF_RNA)
changeCols_2 <- col_names[! col_names %in%
c("CHROM","POS","REF","ALT","Tumor_ID","Master_ID","Patient_ID","GENE","EFFECT","FEATUREID","HGVS_C","SOURCE","geneBiotype","transcrip
tBiotype",
"mutationType","Tumor_entity","Tumor_entity_short","Tumor_state","Metastatic_site","Tumor_origin","Metastasis","Mutation_ID","Biotype_grou
p","Mutation_group")]
all.patients_DF_RNA <- all.patients_DF_RNA[,(changeCols_2):= lapply(.SD, as.numeric), .SDcols = changeCols_2] # change to numeric values

all.patients_DF_RNA <- mutate(all.patients_DF_RNA, Coverage=TumorAD+TumorRD)
all.patients_DF_RNA_filtered <- as.data.table(all.patients_DF_RNA)
#different ways to filter --> select one
all.patients_DF_RNA_filtered <- all.patients_DF_RNA_filtered[TumorVF >=0.05 & Coverage >= 5 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_RNA_filtered <- all.patients_DF_RNA_filtered[Coverage >= 5 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_RNA_filtered <- all.patients_DF_RNA_filtered[TumorVF >=0.05 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_RNA_filtered <- all.patients_DF_RNA_filtered[TumorAD >=2 & NormalAD <= 1]

## plot: compare un-filtered and filtered data sets
ggplot(all.patients_DF_RNA_filtered, aes(x = Tumor_ID)) +
  geom_bar(data = all.patients_DF_RNA, aes(x = Tumor_ID), stat = "count" , fill = "light grey") +
  geom_bar(data = all.patients_DF_RNA_filtered, aes(x = Tumor_ID), stat = "count")+
  #scale_y_continuous(limits = c(0,1.1*max(mutational_load.permaster_bothtools$N)), breaks =
seq(0,1.1*max(mutational_load.permaster_bothtools$N), by = 10000))+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")

all.patients_DF_RNA_filtered[, .N, by=.(Tumor_ID)]

###__ 2c ### Mutational pattern Somatic Mutations _____________________________________________________________#####
all.patients_DF_DNA <- subset(all.patients_DF_uniqueMut, Mutation_group == "Somatic")
all.patients_DF_DNA_subsonly <- all.patients_DF_DNA[all.patients_DF_DNA$mutationType == "substitution",]
all.patients_DF_DNA_subsonly$Ref_Alt <-  paste(all.patients_DF_DNA_subsonly$REF, all.patients_DF_DNA_subsonly$ALT, sep ="_")
all.patients_DF_DNA_subsonly$Ref_Alt_coding <- all.patients_DF_DNA_subsonly$HGVS_C
all.patients_DF_DNA_subsonly$Ref_Alt_coding <- gsub(".*(.)>(.)", "\\1_\\2", all.patients_DF_DNA_subsonly$Ref_Alt_coding)

ggplot(all.patients_DF_DNA_subsonly, aes(x=Ref_Alt_coding, fill = SOURCE))+ #[!all.patients_DF_RNA_small_subsonly$Biotype_group == "Protein
Coding",]
  geom_bar(stat = "count")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  labs(x="Nucleic acid changes Ref_Alt", y="Number of DNA substitutions", fill = "Mutation origin")+
  #scale_fill_manual(labels = c("RNA"), values = c("#619CFF"))+
  #facet_wrap(~ Biotype_group)+ #~
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 25, vjust = 0.8),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")

### 3 ### ::mutationalLoad ____________________________________________________________________________#####
###__ 3a ### ::mutationalLoad:: ::peptideLoad:: ::pep.per.mut:: --- per Patient_ID _____________________________#####
mutational_load <- all.patients_DF_uniqueMut %>%
  distinct(Patient_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Patient_ID) %>%
  summarize(mut_load=n(), Tumor_entity=first(Tumor_entity), mean_TumorVF=mean(TumorVF))

mutational_load_detail <- all.patients_DF_uniqueMut %>%
  distinct(Patient_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Patient_ID) %>%
  mutate(N.mut=n()) %>%
  ungroup() %>%
  group_by(Patient_ID, mutationType) %>%
  summarise(N_mut_type=n(), Tumor_entity=Tumor_entity %>% unique %>% sort %>% paste(collapse = ", "), N.mut=first(N.mut))
###
mutational_load_detail.help <- mutational_load_detail %>%
  group_by(Patient_ID) %>%
  mutate(N.mut=first(N.mut))
###
mutational_load_detail.help["test"] <- duplicated(str_c(mutational_load_detail.help$Tumor_entity, mutational_load_detail.help$Patient_ID))
mutational_load_detail.help <- mutational_load_detail.help %>%
  mutate(Tumor_entity_plot=ifelse(test==T, NA, as.character(Tumor_entity)))
mutational_load_detail.help.test <- mutational_load_detail.help %>%
  mutate(Tumor_entity_plot=ifelse(is.na(Tumor_entity_plot),"",Tumor_entity_plot)) %>%
  group_by(Patient_ID) %>%
  summarise(Tumor_entity_plot_new=paste0(Tumor_entity_plot,collapse = "  "))
mutational_load_detail.help <- merge(mutational_load_detail.help, mutational_load_detail.help.test)
mutational_load_detail.help["test"] <- duplicated(str_c(mutational_load_detail.help$Tumor_entity_plot_new,
mutational_load_detail.help$Patient_ID))
mutational_load_detail.help <- mutational_load_detail.help %>%
  mutate(Tumor_entity_plot=ifelse(test==T, NA, as.character(Tumor_entity_plot_new))) %>%
  select(-test)
###
mutational_load_detail_calledby <- all.patients_DF_uniqueMut %>%
  distinct(Tumor_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Tumor_ID) %>%
  mutate(N.mut=n()) %>%
  ungroup() %>%
  group_by(Tumor_ID, SOURCE) %>%
  summarise(N_mut_type=n(), Tumor_entity=first(Tumor_entity), N.mut=first(N.mut))
###
mutational_load_detail.help_calledby <- mutational_load_detail_calledby %>%
  group_by(Tumor_ID) %>%
  mutate(N.mut=first(N.mut))
###
mutational_load_detail.help_calledby["test"] <- duplicated(str_c(mutational_load_detail.help_calledby$Tumor_entity,
mutational_load_detail.help_calledby$Tumor_ID))
mutational_load_detail.help_calledby <- mutational_load_detail.help_calledby %>%
  mutate(Tumor_entity_plot=ifelse(test==T, NA, as.character(Tumor_entity)))
mutational_load_detail.help_calledby.test <- mutational_load_detail.help_calledby %>%
  mutate(Tumor_entity_plot=ifelse(is.na(Tumor_entity_plot),"",Tumor_entity_plot)) %>%
  group_by(Tumor_ID) %>%
  summarise(Tumor_entity_plot_new=paste0(Tumor_entity_plot,collapse = "  "))
mutational_load_detail.help_calledby <- merge(mutational_load_detail.help_calledby, mutational_load_detail.help_calledby.test)
mutational_load_detail.help_calledby["test"] <- duplicated(str_c(mutational_load_detail.help_calledby$Tumor_entity_plot_new,
mutational_load_detail.help_calledby$Tumor_ID))
mutational_load_detail.help_calledby <- mutational_load_detail.help_calledby %>%
  mutate(Tumor_entity_plot=ifelse(test==T, NA, as.character(Tumor_entity_plot_new))) %>%
  select(-test)

###_____ 3a ____ Plots ___________________________________________________________________________________#####
ggplot(mutational_load_detail[!mutational_load_detail$Patient_ID == "Mel15",], aes(x=Patient_ID, y=N_mut_type, fill=mutationType))+
  geom_bar(position = "stack", stat = "identity")+
  #scale_y_continuous(limits = c(0, 1.1*max(mutational_load_detail$N_mut_type)), breaks = seq(0, 1.1*max(mutational_load_detail$N_mut_type),
by = 10000))+
  #geom_text(aes(label=Tumor_entity, y=(max(mutational_load_detail$N_mut_type)/5)), angle=90, size=6, color=c("#333333"), fontface="plain",
family = "sans", vjust=0)+
  #expand_limits(x=35)+
  #geom_text(aes(label=mutational_load_detail.help$Tumor_entity_plot,
y=(mutational_load_detail.help$N.mut+0.05*max(mutational_load_detail.help$N.mut))), nudge_x = -0.15, angle=35, size=6, color=c("#555555"),
fontface="plain", family = "sans", hjust=0)+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 35))+
  labs(x="Patient ID", y="Number of unique mutations", fill="Mutation Type")

# 3a - 2 ## Mutational load :: Called.By --- per Patient_ID
ggplot(mutational_load_detail_calledby, aes(x=reorder(Patient_ID, desc(N.mut)), y=N_mut_type, fill=SOURCE))+
  geom_bar(position = "stack", stat = "identity")+
  scale_y_continuous(limits = c(0, 1.2*max(mutational_load_detail_calledby$N_mut_type)), breaks = seq(0,
1.2*max(mutational_load_detail_calledby$N_mut_type), by = 10000))+
  expand_limits(x=35)+
  geom_text(aes(label=mutational_load_detail.help_calledby$Tumor_entity_plot,
y=(mutational_load_detail.help_calledby$N.mut+0.05*max(mutational_load_detail.help_calledby$N.mut))), nudge_x = -0.15, angle=35, size=6,
color=c("#555555"), fontface="plain", family = "sans", hjust=0)+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 35))+
  labs(x="Patient ID", y="# of mutations", fill="Called by")

###__ 3b ### ::mutationalLoad:: ::peptideLoad:: ::pep.per.mut:: --- per Master_ID _______________________________#####
mutational_load.permaster <- all.patients_DF_new %>%
  distinct(Master_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Master_ID) %>%
  summarize(mut_load=n(), Tumor_entity=first(Tumor_entity), mean_TumorVF=mean(TumorVF), Patient_ID=first(Patient_ID),
Metastasis=first(Metastasis))

mutational_load_detail.permaster <- all.patients_DF_new %>%
  distinct(Master_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Master_ID) %>%
  mutate(N.mut=n()) %>%
  ungroup() %>%
  group_by(Master_ID) %>%
  summarise(N_mut_type=n(), Tumor_entity=first(Tumor_entity), N.mut=first(N.mut), Patient_ID=first(Patient_ID), Metastasis=first(Metastasis))
%>%
  mutate(Tumor_entity_plot= ifelse(Metastasis=="T1", Tumor_entity, NA)) %>%
  group_by(Patient_ID) %>%
  mutate(N.mut.max=max(N.mut))

## differentiate Exome and RNA called mutations !!!! Old data set, not up to date!!!
### DNA EXOME
mutational_load.permaster_DNA <- all.patients_DF_DNA %>%
  distinct(Master_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Master_ID) %>%
  summarize(mut_load=n(), Tumor_entity=first(Tumor_entity), mean_TumorVF=mean(TumorVF), Patient_ID=first(Patient_ID),
Metastasis=first(Metastasis))

mutational_load_detail.permaster_DNA <- all.patients_DF_DNA %>%
  distinct(Master_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Master_ID) %>%
  mutate(N.mut=n()) %>%
  ungroup() %>%
  group_by(Master_ID) %>%
  summarise(N_mut_type=n(), Tumor_entity=first(Tumor_entity), N.mut=first(N.mut), Patient_ID=first(Patient_ID), Metastasis=first(Metastasis))
%>%
  mutate(Tumor_entity_plot= ifelse(Metastasis=="T1", Tumor_entity, NA)) %>%
  group_by(Patient_ID) %>%
  mutate(N.mut.max=max(N.mut))

### RNA SEQ
```

```
mutational_load.permaster_RNA <- all.patients_DF_RNA %>%
  distinct(Master_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Master_ID) %>%
  summarize(mut_load=n(), Tumor_entity=first(Tumor_entity), mean_TumorVF=mean(TumorVF), Patient_ID=first(Patient_ID),
Metastasis=first(Metastasis))

mutational_load_detail.permaster_RNA <- all.patients_DF_RNA %>%
  distinct(Master_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  group_by(Master_ID) %>%
  mutate(N.mut=n()) %>%
  ungroup() %>%
  group_by(Master_ID) %>%
  summarise(N_mut_type=n(), Tumor_entity=first(Tumor_entity), N.mut=first(N.mut), Patient_ID=first(Patient_ID), Metastasis=first(Metastasis))
%>%
  mutate(Tumor_entity_plot= ifelse(Metastasis=="T1", Tumor_entity, NA)) %>%
  group_by(Patient_ID) %>%
  mutate(N.mut.max=max(N.mut))

## both tools overlapp
mutational_load.permaster_bothtools <- all.patients_DF_new %>%
  distinct(CHROM,POS,REF,ALT,SOURCE,Tumor_ID, .keep_all = T) %>%
  group_by(CHROM,POS,REF,ALT,Tumor_ID) %>%
  summarise(SOURCE=SOURCE %>% unique %>% sort %>% paste(collapse = " + "), Tumor_entity=first(Tumor_entity),Patient_ID=first(Patient_ID),
Metastasis=first(Metastasis)) %>%
  ungroup() %>%
  group_by(Tumor_ID,SOURCE) %>%
  summarise(N=n(), Tumor_entity=first(Tumor_entity), Patient_ID=first(Patient_ID), Metastasis=first(Metastasis))

###_____ 3b ___ Plots (final) ____________________________________________________________#####
# 3b - 9 ## Mutational load --- per Tumor_ID --- Mutation Source -- sorted by ID -- faceted by MutationGroup
all.patients_DF_uniqueMut_Core$Patient_ID <- sub("_", "-", all.patients_DF_uniqueMut_Core$Patient_ID)

ggplot(all.patients_DF_uniqueMut_Core[!all.patients_DF_uniqueMut_Core$Patient_ID == "Mel15",], aes(x = Patient_ID, fill = SOURCE)) +
  geom_bar() +
  #scale_y_continuous(limits = c(0,1.1*max(mutational_load.permaster_bothtools$N)), breaks =
seq(0,1.1*max(mutational_load.permaster_bothtools$N), by = 10000))+
  #scale_fill_brewer(palette = "Set3")+ # or use "Blues"/"Set3"
  theme_PS()+
  facet_wrap(~factor(Mutation_group, levels=c('Somatic','RNA editing')),labeller = as_labeller(c( 'Somatic' = "Somatic mutations", 'RNA editing' =
"RNA alterations")), ncol = 1, scales="free_y")+ #strip.position = "left"
  theme(
    axis.text.x = element_text(angle = 45, size = 20, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 20, face = "bold"),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")+
  #guides (fill = guide_legend(ncol = 1))+
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA"))+
  labs(x="Patient ID", y="Number of unique variants", fill="Mutation Origin")

### duplicate DNA+RNA mutations and group to somatic AND RNA editing to have them in both facets
ggplot(all.patients_DF_uniqueMut_dupliDNARNA[!all.patients_DF_uniqueMut_dupliDNARNA$Patient_ID == "Mel15",], aes(x = Tumor_ID, fill =
SOURCE)) +
  geom_bar() +
  #scale_y_continuous(limits = c(0,1.1*max(mutational_load.permaster_bothtools$N)), breaks =
seq(0,1.1*max(mutational_load.permaster_bothtools$N), by = 10000))+
  #scale_fill_brewer(palette = "Set3")+ # or use "Blues"/"Set3"
  theme_PS()+
  facet_wrap(~factor(Mutation_group, levels=c('Somatic','RNA editing')),labeller = as_labeller(c( 'Somatic' = "Somatic mutations", 'RNA editing' =
"RNA alterations")), ncol = 1, scales="free_y")+ #strip.position = "left"
  theme(
    axis.text.x = element_text(angle = 45, size = 20, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 20, face = "bold"),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")+
  #guides (fill = guide_legend(ncol = 1))+
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA"))+
  labs(x="Patient ID", y="Number of unique variants", fill="Mutation Origin")

###__ 3c ### :: peptides per mutations _____________________________________________________________________#####
#source(file = "Peptides/ImmuNeo_peptides_all_V2.R")
peptide_load <- IN.10 %>%
  group_by(Patient_ID) %>%
  summarise(pep_load=n(), VF.Strelka.mean=mean(TumorVF.StrelkaRNA, na.rm = T), VF.Mutect2.mean=mean(TumorVF.Mutect2, na.rm = T))

peptide.per.mutation <- merge(peptide_load, mutational_load) %>%
  mutate(pep.per.mut=pep_load/mut_load) %>%
  rowwise() %>% mutate(VF.mean= max(VF.Strelka.mean, VF.Mutect2.mean, na.rm = T))

###__ 3d ### :: TMB tumor mutational burder per megabase --- per Master_ID/Tumor_ID _______________________#####
# 3d - 1 ## TMB rescued all --- per Tumor_ID
TMB_resuced <- fread("mutation_calling/Results_export/20220304_TMB_rescued_Niklas.csv")

ggplot(TMB_resuced, aes(x = Sample_ID, y = TMB_probe_rescued)) +
  geom_bar(stat = "identity") +
  #scale_y_continuous(limits = c(0,1.1*max(mutational_load.permaster_bothtools$N)), breaks =
seq(0,1.1*max(mutational_load.permaster_bothtools$N), by = 10000))+
  #scale_fill_brewer(palette = "Set3")+ # or use "Blues"/"Set3"
  theme_PS()+
  #facet_wrap(~factor(Mutation_group, levels=c('Somatic','RNA editing')),labeller = as_labeller(c( 'Somatic' = "Somatic mutations", 'RNA editing' =
"RNA alterations")), ncol = 1, scales="free_y")+ #strip.position = "left"
  theme(
    axis.text.x = element_text(angle = 45, size = 20, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 20, face = "bold"),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")+
  #guides (fill = guide_legend(ncol = 1))+
  #scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA"))+
  labs(x="Patient ID", y="Number of variants per Mb")

### 4 ### :: Mutation Type analysis :: ______________________________________________________#####
###__ 4a ### :: Mutation type - effect - missense, stop loss, etc. :: _______________________________#####
MT <- all.patients_DF_new %>%
  group_by(EFFECT, SOURCE) %>%
  summarise(N.effect_type=n()) %>%
  ungroup()
MT.help <- MT %>%
  group_by(EFFECT) %>%
  summarise(N.effect_type=sum(N.effect_type)) %>%
  top_n(6, N.effect_type)
MT <- all.patients_DF_new %>%
  group_by(EFFECT, SOURCE) %>%
  summarise(N.effect_type=n()) %>%
  filter(EFFECT %in% MT.help$EFFECT) %>%
  ungroup()
MT.temp <- MT %>%
  filter(grepl("Splice", EFFECT)) %>%
  group_by(SOURCE) %>%
  summarise(N.effect_type=sum(N.effect_type), EFFECT="Splice site & intron variant")
MT <- MT %>%
  filter(!grepl("Splice", EFFECT)) %>%
  bind_rows(MT.temp)

## combined unique mutations
## eigther for whole data set "all.patients_DF_uniqueMut" or Core samples "all.patients_DF_uniqueMut_Core" or for subsets "all.patients_DF_DNA"
and "all.patients_DF_RNA"
MT <- all.patients_DF_uniqueMut_Core[!all.patients_DF_uniqueMut_Core$Patient_ID == "Mel15",] %>%
  group_by(EFFECT, SOURCE, Mutation_group) %>%
  summarise(N.effect_type=n()) %>%
  ungroup()
MT.help <- MT %>%
  group_by(EFFECT) %>%
  summarise(N.effect_type=sum(N.effect_type)) %>%
  top_n(6, N.effect_type)
MT <- all.patients_DF_uniqueMut_Core[!all.patients_DF_uniqueMut_Core$Patient_ID == "Mel15",] %>%
  group_by(EFFECT, SOURCE, Mutation_group) %>%
  summarise(N.effect_type=n()) %>%
  filter(EFFECT %in% MT.help$EFFECT) %>%
  ungroup()
MT.temp <- MT %>%
  filter(grepl("Splice", EFFECT)) %>%
  group_by(SOURCE, Mutation_group) %>%
  summarise(N.effect_type=sum(N.effect_type), EFFECT="Splice-site & intron variant")
MT <- MT %>%
  filter(!grepl("Splice", EFFECT)) %>%
  bind_rows(MT.temp)
MT$EFFECT <- sub("transcript exon ","", MT$EFFECT)
MT[MT == "Missense variant"] <- "Coding missense variant"
MT[MT == "Non-coding variant"] <- "Non-coding missense variant"

# for all IN samples
ggplot(MT, aes(x = reorder(EFFECT, desc(-N.effect_type)), y=N.effect_type, fill=SOURCE))+
  geom_bar(position = "stack", stat = "identity")+
  scale_y_continuous(limits = c(0, 1.1*max(MT$N.effect_type)), breaks = seq(0, 1.1*max(MT$N.effect_type), by =10000), labels = comma)+
  theme_PS()+
  coord_flip()+
  labs(x="Mutation effect", y="Number of unique variants", fill="Mutation origin")+
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA")) +
  #scale_y_continuous(labels=comma)+
  theme(legend.position = "bottom",
      axis.text.x = element_text(size= 25),
      axis.text.y = element_text(size= 25),
      plot.title = element_blank(),
      axis.title.y = element_text(size=25),
      axis.title.x = element_text(size=25),
      legend.text = element_text(size= 25),
      legend.title = element_text(size=25, face = "bold"))+
  scale_x_discrete(labels=function(x){sub("\\s", "\n", x)}) #creates axis label breaks

## for core samples with facet warp
ggplot(MT, aes(x = reorder(EFFECT, desc(-N.effect_type)), y=N.effect_type, fill=SOURCE))+
  geom_bar(position = "stack", stat = "identity")+
  #scale_y_continuous(limits = c(0, 1.2*max(MT$N.effect_type)), breaks = seq(0, 1.2*max(MT$N.effect_type), by =10000))+
  theme_PS()+
  scale_y_continuous(labels=comma)+
  coord_flip()+
  labs(x="Mutation effect", y="Number of unique variants", fill="Mutation origin")+
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA")) +
  facet_wrap(~factor(Mutation_group, levels=c('Somatic','RNA editing')),labeller = as_labeller(c( 'Somatic' = "Somatic mutations", 'RNA editing' = 
"RNA alterations")),ncol =1, scales = "free")+
  #facet_grid(~factor(Mutation_group, levels=c('Somatic','RNA editing')),scales = "free")+
  theme(#legend.position = "bottom",
   axis.text.x = element_text(size= 20, hjust=0.8),
   axis.text.y = element_text(size= 20),
   plot.title = element_blank(),
   axis.title.y = element_blank(),
   axis.title.x = element_text(size=25),
   strip.text = element_text(size = 25, face = "bold"),
   #legend.text = element_text(size= 25),
   #legend.title = element_text(size=25, face = "bold"),
   legend.position = "none")+
  scale_x_discrete(labels=function(x){sub("\\s", "\n", x)}) #creates axis label breaks at blank spaces

### 5 ### :: Biotype of mutated Gene :: ______________________________________________#####
# for biotype groups
BT.group.IN <-all.patients_DT_new[!all.patients_DT_new$Patient_ID == "Mel15",]
BT.group.IN[Biotype_group %in% c("regulatory RNAs (antisense)", "lincRNA"), Biotype_group := "regulatory RNAs"]
BT.group.IN[Biotype_group %in% c("regulatory RNAs"), Biotype_group := "Regulatory RNA"]

BT.group.help <- BT.group.IN %>%
  group_by(Patient_ID, CHROM, POS, REF, ALT) %>%
  summarise(Biotype_group=max(Biotype_group), SOURCE=paste0(sort(unique(SOURCE)), collapse = "+")) %>%
  ungroup() %>%
  group_by(Biotype_group) %>%
  summarise(N.gene_biotype=n())

BT.group.IN <- BT.group.IN %>%
  group_by(Patient_ID, CHROM, POS, REF, ALT) %>%
  summarise(Biotype_group=max(Biotype_group), SOURCE=paste0(sort(unique(SOURCE)), collapse = "+")) %>%
  ungroup() %>%
  group_by(Biotype_group, SOURCE) %>%
  summarise(N.gene_biotype=n()) %>%
  filter(Biotype_group %in% BT.group.help$Biotype_group)
```

```
BT.group.summary <- all.patients_DT_uniqueMut
BT.group.summary[grep(";", Biotype_group, ignore.case = T), Biotype_group := "Multiple"]
BT.group.summary <- as.data.frame(BT.group.summary)

## Only core samples with facet wrap
BT.group.summary <- all.patients_DT_uniqueMut_Core
BT.group.summary[grep(";", Biotype_group, ignore.case = T), Biotype_group := "Others"]
BT.group.summary <- as.data.frame(BT.group.summary)

### plots
ggplot(BT.group.IN, aes(x = reorder(Biotype_group, N.gene_biotype), y=N.gene_biotype, fill=SOURCE))+
  geom_bar(position = "stack", stat = "identity")+
  #scale_y_continuous(limits = c(0,1.2*max(BT.group$N.gene_biotype)), breaks =
seq(0,1.2*max(BT.group$N.gene_biotype),round(1.2*max(BT.group$N.gene_biotype)/10, digits = -4)))+
  scale_y_continuous(labels=comma)+
  theme_PS()+
  coord_flip()+
  labs(x="Gene Biotype", y="Number of unique variants", fill="Mutation origin")+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size= 25),
        axis.text.y = element_text(size= 25),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size= 25),
        legend.title = element_text(size=25, face = "bold"))+
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA"))
#scale_fill_brewer(palette = "Paired", direction = -1)

# facet wrap MutationGroup
ggplot(BT.group.summary, aes(x = fct_rev(fct_infreq(Biotype_group)), fill=SOURCE))+
  geom_bar(position = "stack")+
  scale_y_continuous(labels=comma)+
  theme_PS()+
  facet_wrap(~factor(Mutation_group, levels=c('Somatic','RNA editing')), labeller = as_labeller(c( 'Somatic' = "Somatic mutations", 'RNA editing' =
"RNA alterations")), ncol = 1, scales = "free")+   #scales = "free"
  #facet_grid(~factor(Mutation_group, levels=c('Somatic','RNA editing')),scales = "free")+
  coord_flip()+
  labs(x="Gene Biotype", y="Number of unique variants", fill="Mutation origin")+
  theme(#legend.position = "bottom",
    axis.text.x = element_text(size= 20,hjust=0.8),
    axis.text.y = element_text(size= 20),
    plot.title = element_blank(),
    axis.title.y = element_blank(),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 25, face = "bold"),
    #legend.text = element_text(size= 25),
    #legend.title = element_text(size=25, face = "bold"),
    legend.position = "none")+
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA"))
#scale_fill_brewer(palette = "Paired", direction = -1)

### 6 ### :: Mutation Overlap UpSet all patients :: _____________________________________________________#####
###__ 6a #### :: for Mutations _________________________________________________________________________#######
overlap.mutations.RNA <- all.patients_DF_RNA[all.patients_DF_RNA$Patient_ID %in% c("IN_17","IN_19","IN_23","IN_24","IN_27"),] %>% #
all.patients_DF_RNA[all.patients_DF_RNA$Patient_ID %in% c("IN_11","IN_17","IN_19", "IN_23", "IN_24", "IN_27"),] or
all.patients_DF_RNA[!all.patients_DF_RNA$Patient_ID == "Mel15",]
  mutate(Mutation_ID_temp=Mutation_ID) %>%
  select(Mutation_ID, Mutation_ID_temp, GENE, geneBiotype, SOURCE, Tumor_ID) %>%
  mutate_if(is.factor, as.character) %>%
  distinct(Tumor_ID, Mutation_ID_temp, .keep_all = T) %>%
  group_by(Tumor_ID) %>%
  mutate(grouped_id = row_number()) %>%
  spread(Tumor_ID, Mutation_ID_temp) %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  mutate_at(c(6:ncol(.)), ~replace(., .!=0, 1)) %>%
  as.data.frame() %>%
  mutate_at(vars(c(6:ncol(.))), as.numeric) %>%
  #select(-Id, -gene_symbol, -gene_biotype, -grouped_id) %>%
  group_by(Mutation_ID) %>%
  summarise_all(list(~ max(.))) %>% # depricated: summarise_all(funs(max))
  as.data.frame()

overlap.mutations.DNA <- all.patients_DF_DNA[all.patients_DF_DNA$Patient_ID %in% c("IN_11","IN_17","IN_19", "IN_23", "IN_24", "IN_27"),] %>%
# all.patients_DF_DNA[all.patients_DF_DNA$Patient_ID %in% c("IN_11","IN_17","IN_19", "IN_23", "IN_24", "IN_27"),] or
all.patients_DF_DNA[!all.patients_DF_DNA$Patient_ID == "Mel15",]
```

```
 mutate(Mutation_ID_temp=Mutation_ID) %>%
 select(Mutation_ID, Mutation_ID_temp, GENE, geneBiotype, SOURCE, Tumor_ID) %>%
 mutate_if(is.factor, as.character) %>%
 distinct(Tumor_ID, Mutation_ID_temp, .keep_all = T) %>%
 group_by(Tumor_ID) %>%
 mutate(grouped_id = row_number()) %>%
 spread(Tumor_ID, Mutation_ID_temp) %>%
 mutate_all(~replace(., is.na(.), 0)) %>%
 mutate_at(c(6:ncol(.)), ~replace(., .!=0, 1)) %>%
 as.data.frame() %>%
 mutate_at(vars(c(6:ncol(.))), as.numeric) %>%
 #select(-Id, -gene_symbol, -gene_biotype, -grouped_id) %>%
 group_by(Mutation_ID) %>%
 summarise_all(list(~ max(.))) %>% # depricated: summarise_all(funs(max))
 as.data.frame()

###_____ 6a ___ Plots For mutations _________________________________________________________________ ######
### for general overlapp of mutations
colnames(overlap.mutations.RNA) <- gsub("_", "-", colnames(overlap.mutations.RNA))
# with complex upset
Patients <- colnames(overlap.mutations.RNA)[-1:-5]
#general overview
upset(overlap.mutations, Patients,mode = 'inclusive_intersection', base_annotations=list('Intersection
size'=intersection_size(counts=FALSE,mapping=aes(fill=GENE))+ theme(legend.position = "none")), width_ratio = 0.1,height_ratio = 1,
n_intersections =100)

#final overview plot --> save as pdf, device size, 15x20
upset(overlap.mutations.RNA, Patients, base_annotations=list('Intersection size'=intersection_size(counts=FALSE,mapping=aes(fill=GENE))+
theme(legend.position = "none")), width_ratio = 0.2,height_ratio = 1.7, n_intersections =100,sort_sets=FALSE,
themes=upset_default_themes(text=element_text(size=20)))
upset(overlap.mutations.DNA, Patients, base_annotations=list('Intersection size'=intersection_size(counts=FALSE,mapping=aes(fill=GENE))+
theme(legend.position = "none")), width_ratio = 0.2,height_ratio = 1.7, n_intersections =50,sort_sets=FALSE,
themes=upset_default_themes(text=element_text(size=20)))

# final subset plot
upset(overlap.mutations, Patients, mode = 'exclusive_intersection',set_sizes=FALSE, base_annotations=list('Intersection
size'=intersection_size(counts=TRUE,mapping=aes(fill=GENE))+ theme(legend.position = "none")),min_size = 2, min_degree = 10, width_ratio =
0.1,keep_empty_groups=TRUE, height_ratio = 1, n_intersections =100,
themes=upset_default_themes(text=element_text(size=20)),sort_sets=FALSE)

# only multi-tumor patients --> compare metastases
Patients <- colnames(overlap.mutations.RNA)[-1:-5]
upset(overlap.mutations.RNA, Patients, base_annotations=list('Intersection size'=intersection_size(counts=TRUE,text_mapping =
aes(label=paste0(round(!!get_size_mode('exclusive_intersection')/!!get_size_mode('inclusive_union') * 100), '%')))+ theme(legend.position =
"none")), width_ratio = 0.2,height_ratio = 1.7,min_degree=2, max_degree = 3,min_size = 42, sort_sets=FALSE,
themes=upset_default_themes(text=element_text(size=20)))

Patients <- colnames(overlap.mutations.DNA)[-1:-5]
upset(overlap.mutations.DNA, Patients, base_annotations=list('Intersection size'=intersection_size(counts=TRUE,text_mapping =
aes(label=paste0(round(!!get_size_mode('exclusive_intersection')/!!get_size_mode('inclusive_union') * 100), '%')))+ theme(legend.position =
"none")), width_ratio = 0.2,height_ratio = 1.7,min_degree=2, max_degree = 3,min_size = 4, sort_sets=FALSE,
themes=upset_default_themes(text=element_text(size=20)))

upset(overlap.mutations, Patients, base_annotations=list('Intersection size'=intersection_size(counts=TRUE,mapping=aes(fill=GENE))),min_size=
5,max_size = 7, min_degree = 7, width_ratio = 0.1,height_ratio = 1,keep_empty_groups=TRUE, n_intersections =10)

# for mutations filtered by genes in Top20 Hotspot genes
Top_mut_genes <- as.vector(unique(geneDF.overlap$GENE))
overlap.mutations.Top20 <- subset(overlap.mutations, GENE %in% Top_mut_genes)

upset(overlap.mutations.Top20, Patients, base_annotations=list('Intersection size'=intersection_size(counts=TRUE,mapping=aes(fill=GENE))+
theme(legend.position = "none")), min_degree = 3, width_ratio = 0.1,height_ratio = 1, n_intersections =100)

###___ 6a.2 ##### :: summary matrix overlap mutations DNA & RNA _________________________________________#######
# create column with all samples where mutation is present
overlap.mutations <- as.data.table(overlap.mutations)
dt <- overlap.mutations[, -2:-5]
patnames <- gsub(" ", "",as.data.table(melt(dt, "Mutation_ID"))[,toString(variable[value==1]), Mutation_ID]$V1)
overlap.mutations[, Samples := patnames]
setcolorder(overlap.mutations, c(colnames(overlap.mutations)[1:5], 'Samples'))

# expand matrix and add perPatient information
dt <- overlap.mutations[, -2:-6]
meltDT <- as.data.table(melt(dt, "Mutation_ID"))
meltDT[, variable := gsub("_T\\d$", "", variable)]
meltDT <- meltDT[, max(value), by=.(Mutation_ID, variable)]
```

```
library(tidyr)
dtnew <- spread(meltDT, variable, V1)
patnames <- gsub(" ", "",as.data.table(melt(dtnew, "Mutation_ID"))[,toString(variable[value==1]), Mutation_ID]$V1)
dtnew[, Patients := patnames]
setcolorder(dtnew, c('Mutation_ID', 'Patients'))

overlap.mutations.RNA.summary <- merge(overlap.mutations[,1:6],dtnew[,1:2])
overlap.mutations.DNA.summary <- merge(overlap.mutations[,1:6],dtnew[,1:2])

# count number of samples/patients
# RNA alterations
sample_count <- lengths(regmatches(overlap.mutations.RNA.summary$Samples, gregexpr(",", overlap.mutations.RNA.summary$Samples)))+1
overlap.mutations.RNA.summary[, count_samples := sample_count]
patient_count <- lengths(regmatches(overlap.mutations.RNA.summary$Patients, gregexpr(",", overlap.mutations.RNA.summary$Patients)))+1
overlap.mutations.RNA.summary[, count_patients := patient_count]
#filtering step if wanted
overlap.mutations.RNA.summary.filtered <- overlap.mutations.RNA.summary[count_patients >= 10]

# modify by hand if needed
overlap.mutations.RNA.summary.filtered <- fread("mutation_calling/Results_export/20211117_shared_RNAMut_filtered_overview_expanded.csv")

# DNA/Somatic mutations
sample_count <- lengths(regmatches(overlap.mutations.DNA.summary$Samples, gregexpr(",", overlap.mutations.DNA.summary$Samples)))+1
overlap.mutations.DNA.summary[, count_samples := sample_count]
patient_count <- lengths(regmatches(overlap.mutations.DNA.summary$Patients, gregexpr(",", overlap.mutations.DNA.summary$Patients)))+1
overlap.mutations.DNA.summary[, count_patients := patient_count]
#filtering step if wanted
overlap.mutations.DNA.summary.filtered <- overlap.mutations.DNA.summary[count_patients >= 4]

overlap.mutations.DNA.counted.summary <- overlap.mutations.DNA.summary[,.N, by = count_patients]
overlap.mutations.RNA.counted.summary <- overlap.mutations.RNA.summary[,.N, by = count_patients]

###_____ 6a.2 ___ Plots For filtered and summarized data ________________________________________________########
# colour genes
# for all Tumor Samples!!!
n <- 47
qual_col_pals = brewer.pal.info[brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(brewer.pal, qual_col_pals$maxcolors, rownames(qual_col_pals)))
ggplot(overlap.mutations.counted.filtered, aes(x= factor(count_samples), fill = GENE))+
  geom_bar(stat = "count", position = "stack")+
  scale_x_discrete(drop = FALSE,limits =c("4", "5", "6", "7", "8","9", "10", "11", "12", "13", "14", "15", "16"))+
  scale_y_continuous(breaks=seq(0, 20, 2))+
  scale_fill_manual(drop = FALSE, values=sample(col_vector, n))+
  labs(x="Number of tumor samples", y="Number of unique mutations")+
  theme(#legend.position = "bottom",
    axis.text.x = element_text(size= 20),
    axis.text.y = element_text(size= 20),
    plot.title = element_blank(),
    axis.title.y =  element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 25, face = "bold"),
    #legend.text = element_text(size= 25),
    #legend.title = element_text(size=25, face = "bold"),
    legend.position = "right")+
  guides(fill=guide_legend(ncol=2))

# for all Patients!!! overlap.mutations.counted.filtered = somatic ; overlap.mutations.RNA.summary.filtered = RNA alterations
n <- 35
qual_col_pals = brewer.pal.info[brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(brewer.pal, qual_col_pals$maxcolors, rownames(qual_col_pals)))
ggplot(overlap.mutations.counted.filtered[overlap.mutations.counted.filtered$count_patients >=4], aes(x= factor(count_patients), fill = GENE))+
  geom_bar(stat = "count", position = "stack")+
  #scale_x_discrete(drop = FALSE,limits =c("4", "5", "6", "7", "8","9", "10", "11", "12", "13", "14"))+
  #scale_y_continuous(breaks=seq(0, 20, 2))+
  #scale_fill_manual(drop = FALSE, values=sample(col_vector, n))+
  scale_fill_brewer(palette = "Set2")+
  #scale_fill_manual(labels = c("DNA + RNA"), values = c("#00BA38"))+
  labs(x="Amount of sharing patients", y="Number of unique somatic mutations shared", fill = "Mutation origin")+
  theme(#legend.position = "bottom",
    axis.text.x = element_text(size= 20),
    axis.text.y = element_text(size= 20),
    plot.title = element_blank(),
    axis.title.y =  element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 25, face = "bold"),
    legend.text = element_text(size= 20),
    legend.title = element_text(size=25, face = "bold"),
    legend.position = "bottom")+
  guides(fill=guide_legend(ncol=1))

# for unfiltered matrix
ggplot(overlap.mutations.RNA.summary[overlap.mutations.RNA.summary$count_patients >=10], aes(x= factor(count_patients), fill = SOURCE))+
  geom_bar(stat = "count", position = "stack")+
  #scale_x_discrete(drop = FALSE,limits =c("4", "5", "6", "7", "8","9", "10", "11", "12", "13", "14"))+
  #scale_y_continuous(breaks=seq(0, 20, 2))+
  scale_fill_manual(labels = c("DNA + RNA", "RNA"), values = c("#00BA38", "#619CFF"))+ #if DNA and RNA alterations included
  #scale_fill_manual(labels = c("RNA"), values = c("#619CFF"))+ # if only RNA alteratiosn included due to filtering
  labs(x="Number of sharing patients", y="Number of unique RNA alterations", fill = "Mutation origin")+
  theme_PS()+
  theme(#legend.position = "bottom",
    axis.text.x = element_text(size= 20),
    axis.text.y = element_text(size= 20),
    plot.title = element_blank(),
    axis.title.y =  element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 25, face = "bold"),
    legend.text = element_text(size= 20),
    legend.title = element_text(size=25, face = "bold"),
    legend.position = "bottom")+
  guides(fill=guide_legend(ncol=2))

ggplot(overlap.mutations.DNA.summary[overlap.mutations.DNA.summary], aes(x= factor(count_patients), fill = SOURCE))+
  geom_bar(stat = "count", position = "stack")+
  #scale_x_discrete(drop = FALSE,limits =c("1", "2", "3", "4", "5", "6", "7", "8","9", "10", "11", "12", "13", "14"))+ #unfiltered, if DNA and DNA+RNA
included
  #scale_x_discrete(drop = FALSE,limits =c("4", "5", "6", "7", "8","9", "10", "11", "12", "13", "14"))+ # filtered, only DNA+RNA mutations included
  #scale_y_continuous(breaks=seq(0, 20, 2))+
  scale_fill_manual(labels = c("DNA", "DNA + RNA"), values = c("#F8766D", "#00BA38"))+ #unfiltered, if DNA and DNA+RNA  included
  #scale_fill_manual(labels = c("DNA + RNA"), values = c("#00BA38"))+ # filtered, only DNA+RNA mutations included
  labs(x="Number of sharing patients", y="Number of unique somatic mutations", fill = "Mutation origin")+
  theme_PS()+
  theme(#legend.position = "bottom",
    axis.text.x = element_text(size= 20),
    axis.text.y = element_text(size= 20),
    plot.title = element_blank(),
    axis.title.y =  element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 25, face = "bold"),
    legend.text = element_text(size= 20),
    legend.title = element_text(size=25, face = "bold"),
    legend.position = "bottom")+
  guides(fill=guide_legend(ncol=2))

###__ 6b #### :: for Genes__________________________________________________________________ ######
overlap.mutations.gene <- all.patients_DF_RNA[!all.patients_DF_RNA$Patient_ID == "Mel15",] %>%
  mutate(Gene_temp=GENE) %>%
  select(Gene_temp, GENE, geneBiotype, SOURCE, Master_ID, Tumor_ID) %>%
  mutate_if(is.factor, as.character) %>%
  distinct(Tumor_ID, Gene_temp, .keep_all = T) %>%
  group_by(Tumor_ID) %>%
  mutate(grouped_id = row_number()) %>%
  spread(Tumor_ID, Gene_temp) %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  mutate_at(c(6:ncol(.)), ~replace(., .!=0, 1)) %>%
  as.data.frame() %>%
  mutate_at(vars(c(6:ncol(.))), as.numeric) %>%
  #select(-Id, -gene_symbol, -gene_biotype, -grouped_id) %>%
  group_by(GENE) %>%
  summarise_all(list(~ max(.))) %>% # depricated: summarise_all(funs(max))
  as.data.frame()

###_____ 6b ___ Plots For mutated Genes ________________________________________________________#######
Patients <- colnames(overlap.mutations.gene)[-1:-5]
#general
upset(overlap.mutations.gene, Patients, base_annotations=list('Intersection size'=intersection_size(counts=FALSE,mapping=aes(fill=GENE))+
theme(legend.position = "none")), width_ratio = 0.1,height_ratio = 1, n_intersections =100)
# filtered
upset(overlap.mutations.gene, Patients, base_annotations=list('Intersection size'=intersection_size(counts=TRUE,mapping=aes(fill=GENE))+
theme(legend.position = "none")), min_degree = 6, min_size = 2,width_ratio = 0.1,height_ratio = 1, n_intersections =100)
upset(overlap.mutations.gene, Patients, base_annotations=list('Intersection size'=intersection_size(counts=TRUE,mapping=aes(fill=GENE))+
theme(legend.position = "none")),max_size =30, min_degree = 10, width_ratio = 0.1,height_ratio = 1, n_intersections =100)
```

```
upset(overlap.mutations.gene, Patients, base_annotations=list('Intersection size'=intersection_size(counts=TRUE,mapping=aes(fill=GENE))+
theme(legend.position = "none")), min_degree = 3, width_ratio = 0.1,height_ratio = 1, n_intersections =100)

###__ 6c #### :: extract list with overlaps (code by Niklas de Andrade-Krätzig, adapted) ______________________ ######
# change into data.table for list generation
overlap.mutations <- as.data.table(overlap.mutations)
overlap.mutations.gene <- as.data.table(overlap.mutations.gene)

# get list of columns
RawMatrix <- overlap.mutations[, -1:-5] # only get data columns !!! check before!! --> if appended matrix use [, -1:-6]
RawList <- list() # init list
for(col in colnames(RawMatrix)){
  seqcol <- "Mutation_ID"
  tmp <- overlap.mutations[, .(get(seqcol), get(col))]
  seqs <- tmp[V2 == 1, V1]
  RawList[[ col ]] <- seqs # append vector to list using name from col
}

# for genes
RawListGenes <- list() # init list
for(col in colnames(RawMatrix)){
  seqcol <- "GENE"
  overlap.mutations <- data.table(overlap.mutations)
  tmp <- overlap.mutations[, .(get(seqcol), get(col))]
  seqs <- tmp[V2 == 1, V1]
  RawListGenes[[ col ]] <- seqs # append vector to list using name from col
}

# source of this function: https://github.com/hms-dbmi/UpSetR/issues/85#issuecomment-327900647
fromList <- function (input) {
  elements <- unique(unlist(input))
  data <- unlist(lapply(input, function(x) {
    x <- as.vector(match(elements, x))
  }))
  data[is.na(data)] <- as.integer(0)
  data[data != 0] <- as.integer(1)
  data <- data.frame(matrix(data, ncol = length(input), byrow = F))
  data <- data[which(rowSums(data) != 0), ]
  names(data) <- names(input)
  row.names(data) <- elements
  return(data)
}
overlapGroups <- function (listInput, sort = TRUE) {
  listInputmat    <- fromList(listInput) == 1
  listInputunique <- unique(listInputmat)
  grouplist <- list()
  for (i in 1:nrow(listInputunique)) {
    currentRow <- listInputunique[i,]
    myelements <- which(apply(listInputmat,1,function(x) all(x == currentRow)))
    attr(myelements, "groups") <- currentRow
    grouplist[[paste(colnames(listInputunique)[currentRow], collapse = ":")]] <- myelements
    myelements
  }
  if (sort) {
    grouplist <- grouplist[order(sapply(grouplist, function(x) length(x)), decreasing = TRUE)]
  }
  attr(grouplist, "elements") <- unique(unlist(listInput))
  return(grouplist)
  # save element list to facilitate access using an index in case rownames are not named
}

li.mut <- overlapGroups(RawList)
saveRDS(li.mut, "Upset_Matrix_RNA_Mutations_LIgroups.rds")

li.mut.gene <- overlapGroups(RawListGenes)
saveRDS(li, "Upset_Matrix_filtered_genes_LIgroups.rds")

### export summary of big matrix, filtered for numbers of patients sharing mutations
### !!! rather use the direct counting from the matrix above !!!
# use list as input
#li.mut.output.somatic <- names(li.mut)
#sample_count <- lengths(regmatches(li.mut.output.somatic, gregexpr(":", li.mut.output.somatic)))+1
#dt.mut.output.somatic <- data.table(names = li.mut.output.somatic, count = sample_count)
#dt.mut.output.somatic.filtered <- dt.mut.output.somatic[count >= 4]
#dt.mut.output.somatic.summary <- dt.mut.output.somatic.filtered[,.N, by = count]
```

```
#li.mut.output.RNA <- names(li.mut)
#sample_count <- lengths(regmatches(li.mut.output.RNA, gregexpr(":", li.mut.output.RNA)))+1
#dt.mut.output.RNA <- data.table(names = li.mut.output.RNA, count = sample_count)
#dt.mut.output.RNA.filtered <- dt.mut.output.somatic[count >= 4]
#dt.mut.output.RNA.summary <- dt.mut.output.somatic.filtered[,.N, by = count]

# use matrix as input #includes Mutation_ID info
#overlap.mutations.counted <- cbind(overlap.mutations[,1:5], data.table(count = rowSums(RawMatrix)))
#overlap.mutations.counted.filtered <- overlap.mutations.counted[count >=4]
#overlap.mutations.counted.summary <- overlap.mutations.counted.filtered[,.N, by = count]

# for RNA rather use the counting from above
#overlap.mutations.RNA.counted <- cbind(overlap.mutations[,1:6], data.table(count = rowSums(RawMatrix)))
#overlap.mutations.RNA.counted.filtered <- overlap.mutations.RNA.counted[count >=4]
#overlap.mutations.RNA.counted.summary <- overlap.mutations.RNA.counted.filtered[,.N, by = count]

write.xlsx(overlap.mutations.counted.filtered,"mutation_calling/Results_export/20211114_shared_somaticMut_filtered_overview.xlsx")
write.xlsx(overlap.mutations.RNA.counted.filtered,"mutation_calling/Results_export/20211114_shared_RNAMut_filtered_overview.xlsx")
write.xlsx(overlap.mutations.RNA.counted.summary,"mutation_calling/Results_export/20211114_shared_RNAMut_filtered_summary_overview.xls
x")
# manually add sample IDs based on li.mut output
# re-load for plotting
overlap.mutations.counted.filtered <- fread("mutation_calling/Results_export/20211114_shared_somaticMut_filtered_overview_expanded.csv")

###__ 6d #### :: extract data table with all shared mutations ______________________________________________ ####
# Ref Table for shared mutations
shared_mutations_DNA_ref <- overlap.mutations.DNA.summary.filtered[,!3:9]
shared_mutations_RNA_ref <- overlap.mutations.RNA.summary.filtered[,!3:9]

shared_mutations_RNA_Upset_ref <- fread("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/27 R scripts Philipp und
Niklas/Philipp/mutation_calling/Results_export/Shared_mutations_RNAall_summary_V2new.csv")
shared_mutations_RNA_Upset_ref <- shared_mutations_RNA_Upset_ref[,c("Mutation_ID", "Gene")]

shared_mutations <- c("1_45694928_T_C", "14_50440012_C_T",
"14_52775636_T_C","15_41163832_T_C","15_41299144_A_G","15_75353745_A_G","15_84641599_T_C","16_81030835_A_G","19_13773270_A_
G","19_13773604_A_G","21_33264079_A_G","15_84641974_T_C","21_33264844_A_G","21_33264854_A_G","21_33264916_A_G","21_33264920
_A_G","12_103939508_G_A","16_21835514_C_A","19_1428841_G_A
","7_128653979_A_G","20_4629698_C_T","20_4629706_G_A","20_4629727_C_T ","14_52775933_T_C","15_41299220_A_G","7_66740100_T_C
","1_120805695_G_C","1_120805696_A_G","1_120805697_G_T","5_98213710_C_G","5_98213711_T_C","20_57488521_C_A","20_57488540_T_C
","7_39834467_G_A","7_39834470_A_C","1_145573538_G_T","1_145573539_C_T","21_6235107_A_T","21_6235109_A_G
","10_50093288_G_A","10_50113947_G_C","3_12606199_A_T","3_12606200_C_G", "1_2395872_G_T","1_2395873_T_G
","1_63508526_A_T","1_63508527_C_T", "14_90404515_G_A","14_90404516_T_A")
shared_mutations <- shared_mutations_ref$Mutation_ID
shared_mutations_DNA <- shared_mutations_DNA_ref$Mutation_ID
shared_mutations_RNA <- shared_mutations_RNA_ref$Mutation_ID
shared_mutations_RNA_Upset <- shared_mutations_RNA_Upset_ref$Mutation_ID

## for somatic shared mutations from general analysis
DF_shared_mutations_DNA <- all.patients_DF_new[!all.patients_DF_new$Patient_ID == "Mel15",]
DF_shared_mutations_DNA <- DF_shared_mutations_DNA[DF_shared_mutations_DNA$Mutation_ID %in% shared_mutations_DNA,]
DF_shared_mutations_DNA <- mutate(DF_shared_mutations_DNA, TumorCoverage=as.numeric(TumorAD)+as.numeric(TumorRD))
DF_shared_mutations_DNA <- transform(DF_shared_mutations_DNA, TumorVF = as.numeric(TumorVF))

DF_shared_mutations_DNA_summary <- aggregate(DF_shared_mutations_DNA[, c('TumorAD', 'TumorRD', 'NormalAD', 'NormalRD', 'TumorVF',
'TumorCoverage')], list(DF_shared_mutations_DNA[, c('Mutation_ID')]), mean)

## for RNA shared alterations from general analysis
DF_shared_mutations_RNA <- all.patients_DF_new[!all.patients_DF_new$Patient_ID == "Mel15",]
DF_shared_mutations_RNA <- DF_shared_mutations_RNA[DF_shared_mutations_RNA$Mutation_ID %in% shared_mutations_RNA,]
DF_shared_mutations_RNA <- mutate(DF_shared_mutations_RNA, TumorCoverage=as.numeric(TumorAD)+as.numeric(TumorRD))
DF_shared_mutations_RNA <- transform(DF_shared_mutations_RNA, TumorVF = as.numeric(TumorVF))

DF_shared_mutations_RNA_summary <- aggregate(DF_shared_mutations_RNA[, c('TumorAD', 'TumorRD', 'NormalAD', 'NormalRD', 'TumorVF',
'TumorCoverage')], list(DF_shared_mutations_RNA[, c('Mutation_ID')]), mean)

## for RNA shared alterations from Upset plot/analysis
DF_shared_mutations_RNA_upset <- all.patients_DF_new[!all.patients_DF_new$Patient_ID == "Mel15",]
DF_shared_mutations_RNA_upset <- DF_shared_mutations_RNA_upset[DF_shared_mutations_RNA_upset$Mutation_ID %in%
shared_mutations_RNA_Upset,]
DF_shared_mutations_RNA_upset <- mutate(DF_shared_mutations_RNA_upset, TumorCoverage=as.numeric(TumorAD)+as.numeric(TumorRD))
DF_shared_mutations_RNA_upset <- transform(DF_shared_mutations_RNA_upset, TumorVF = as.numeric(TumorVF))

DF_shared_mutations_RNA_upset_summary <- aggregate(DF_shared_mutations_RNA_upset[, c('TumorAD', 'TumorRD', 'NormalAD', 'NormalRD',
'TumorVF', 'TumorCoverage')], list(DF_shared_mutations_RNA_upset[, c('Mutation_ID')]), mean)

### 7 ### :: RNA editing evaluations: Ref-Alt, VF and Coverage check & filtering :: ______________________________#####
```

```
###__ 7a #### :: filter RNAonly mut for general criteria (allel freq. and coverage) ___________________________##########
all.patients_DF_RNA <- subset(all.patients_DF_uniqueMut[!all.patients_DF_uniqueMut$Patient_ID == "Mel15",], Mutation_group == "RNA editing")
colnames(all.patients_DF_RNA)
Cols <-
c("CHROM","POS","REF","ALT","Tumor_ID","Master_ID","Patient_ID","GENE","EFFECT","FEATUREID","HGVS_C","TumorVF","TumorAD","TumorRD",
"NormalAD","NormalRD","SOURCE","geneBiotype","transcriptBiotype", "mutationType",
"NormalVF","Tumor_entity","Tumor_entity_short","Tumor_state","Metastatic_site","Tumor_origin","Metastasis","Mutation_ID","Biotype_group","
Mutation_group")

all.patients_DF_RNA_small<- all.patients_DF_RNA[ ,which((names(all.patients_DF_RNA) %in% Cols)==TRUE)]
all.patients_DF_RNA_small <- as.data.table(all.patients_DF_RNA_small)
col_names <- colnames(all.patients_DF_RNA_small)
changeCols_2 <- col_names[! col_names %in%
c("CHROM","POS","REF","ALT","Tumor_ID","Master_ID","Patient_ID","GENE","EFFECT","FEATUREID","HGVS_C","SOURCE","geneBiotype","transcrip
tBiotype",
"mutationType","Tumor_entity","Tumor_entity_short","Tumor_state","Metastatic_site","Tumor_origin","Metastasis","Mutation_ID","Biotype_grou
p","Mutation_group")]
all.patients_DF_RNA_small <- all.patients_DF_RNA_small[,(changeCols_2):= lapply(.SD, as.numeric), .SDcols = changeCols_2] # change to numeric
values

all.patients_DF_RNA_small <- mutate(all.patients_DF_RNA_small, Coverage=TumorAD+TumorRD)
all.patients_DF_RNA_small_filtered <- as.data.table(all.patients_DF_RNA_small)
all.patients_DF_RNA_small_filtered <- all.patients_DF_RNA_small_filtered[TumorVF >=0.05 & Coverage >= 5 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_RNA_small_filtered <- all.patients_DF_RNA_small_filtered[Coverage >= 5 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_RNA_small_filtered <- all.patients_DF_RNA_small_filtered[TumorVF >=0.05 & TumorAD >=2 & NormalAD <= 1]
all.patients_DF_RNA_small_filtered <- all.patients_DF_RNA_small_filtered[TumorAD >=2 & NormalAD <= 1]

mutations_counts_RNAediting_filtered <- all.patients_DF_RNA_small_filtered[, .N, by=.(Tumor_ID)]

## plot: compare un-filtered and filtered data sets
ggplot(all.patients_DF_RNA, aes(x = Tumor_ID)) +
  geom_bar(data = all.patients_DF_RNA, aes(x = Tumor_ID), stat = "count" , fill = "light grey") +
  geom_bar(data = all.patients_DF_RNA_small_filtered, aes(x = Tumor_ID), stat = "count")+
  #scale_y_continuous(limits = c(0,1.1*max(mutational_load.permaster_bothtools$N)), breaks =
seq(0,1.1*max(mutational_load.permaster_bothtools$N), by = 10000))+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")

###__ 7b #### :: Evaluate Ref_Alt, check wt DNA coverage, check A->G pattern for substitutions only _________##########
all.patients_DF_RNA_small$Ref_Alt_coding <- all.patients_DF_RNA_small$HGVS_C

#only look at substitutions (no InDels)
all.patients_DF_RNA_small_subsonly <- all.patients_DF_RNA_small[all.patients_DF_RNA_small$mutationType == "substitution",]
#all.patients_DF_RNA_small_subsonly <- all.patients_DF_RNA_small_filtered[all.patients_DF_RNA_small_filtered$mutationType == "substitution",]

all.patients_DF_RNA_small_subsonly$Ref_Alt_coding <- gsub(".*(.)>(.)", "\\1_\\2", all.patients_DF_RNA_small_subsonly$Ref_Alt_coding)

#merge DNA coverage info
setDT(all.patients_DF_RNA_small_subsonly)[ , DNACoverage :=
all.patients_DF_uniqueMut_DNAcoverage$TumorCoverage.Mutect2.filtered[match(all.patients_DF_RNA_small_subsonly$Mutation_ID ,
all.patients_DF_uniqueMut_DNAcoverage$Mutation_ID)] , ]
all.patients_DF_RNA_small_subsonly$DNACoverage_short <- all.patients_DF_RNA_small_subsonly$DNACoverage
all.patients_DF_RNA_small_subsonly$DNACoverage_short[all.patients_DF_RNA_small_subsonly$DNACoverage_short < 3] <- "no"
all.patients_DF_RNA_small_subsonly$DNACoverage_short[all.patients_DF_RNA_small_subsonly$DNACoverage >= 3] <- "yes"
all.patients_DF_RNA_small_subsonly$Biotype_group[all.patients_DF_RNA_small_subsonly$Biotype_group != "Protein Coding"] <- "Non-coding"

all.patients_DF_RNA_small_subsonly[, .N, by=.(DNACoverage_short)]

## for all RNA only variants from original data set
all.patients_DF_uniqueMut_DNAcoverage_RNAonly <-
all.patients_DF_uniqueMut_DNAcoverage[all.patients_DF_uniqueMut_DNAcoverage$SOURCE == "StrelkaRNA",]
#all.patients_DF_uniqueMut_DNAcoverage_RNAonly <-
all.patients_DF_uniqueMut_DNAcoverage_RNAonly[all.patients_DF_uniqueMut_DNAcoverage_RNAonly$mutationType == "substitution",]
all.patients_DF_uniqueMut_DNAcoverage_RNAonly$DNACoverage_short <-
all.patients_DF_uniqueMut_DNAcoverage_RNAonly$TumorCoverage.Mutect2.filtered
all.patients_DF_uniqueMut_DNAcoverage_RNAonly$DNACoverage_short[all.patients_DF_uniqueMut_DNAcoverage_RNAonly$DNACoverage_short
< 3] <- "no"
```

```
all.patients_DF_uniqueMut_DNAcoverage_RNAonly$DNACoverage_short[all.patients_DF_uniqueMut_DNAcoverage_RNAonly$TumorCoverage.Mut
ect2.filtered >= 3] <- "yes"

all.patients_DF_uniqueMut_DNAcoverage_RNAonly[, .N, by=.(DNACoverage_short)]

# plot
ggplot(all.patients_DF_RNA_small_subsonly, aes(x=Ref_Alt_coding, fill = DNACoverage_short))+
#[!all.patients_DF_RNA_small_subsonly$Biotype_group == "Protein Coding",]
  geom_bar(stat = "count")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  labs(x="Nucleic acid changes Ref_Alt", y="Number of RNA substitutions", fill = "Coverage on DNA above 5 reads")+
  #scale_fill_manual(labels = c("RNA"), values = c("#619CFF"))+
  #facet_wrap(~ Biotype_group)+ #~
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    strip.text.x = element_text(size = 25, face = "bold"),
    axis.text.y = element_text(size = 25, vjust = 0.8),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")
all.patients_DF_RNA_small_subsonly[, .N, by=.(Ref_Alt_coding)]

nb.cols <- 12
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.cols)
ggplot(all.patients_DF_RNA_small_subsonly, aes(x="", fill = Ref_Alt_coding))+
  geom_bar(stat = "count", width = 1)+
  coord_polar("y", start=0)+
  scale_fill_manual(values = mycolors)+
  theme_minimal()+
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.border = element_blank(),
    panel.grid=element_blank(),
    axis.ticks = element_blank(),
    plot.title=element_text(size=14, face="bold")
  )
all.patients_DF_RNA_small_subsonly[, .N, by=.(Ref_Alt_coding)]

###__ 7c #### :: check RNAediting against other data bases _____________________________________________##########
#setDT(all.patients_DF_RNA_small_subsonly)[ , RNAedit.RADAR :=
all.patients_DF_uniqueMut_DNAcoverage$RNAedit.RADAR[match(all.patients_DF_RNA_small_subsonly$Mutation_ID ,
all.patients_DF_uniqueMut_DNAcoverage$Mutation_ID)] , ]
setDT(all.patients_DF_RNA_small_subsonly)[ , RNAedit.REDI :=
all.patients_DF_uniqueMut_DNAcoverage$RNAedit.REDI[match(all.patients_DF_RNA_small_subsonly$Mutation_ID ,
all.patients_DF_uniqueMut_DNAcoverage$Mutation_ID)] , ]

#all.patients_DF_RNA_small_subsonly$Database <- paste(all.patients_DF_RNA_small_subsonly$RNAedit.RADAR,
all.patients_DF_RNA_small_subsonly$RNAedit.REDI, sep = "_")
#all.patients_DF_RNA_small_subsonly[Database %in% c("yes_yes"), Database := "both"]
#all.patients_DF_RNA_small_subsonly[Database %in% c("no_no"), Database := "none"]
#all.patients_DF_RNA_small_subsonly[Database %in% c("no_yes"), Database := "RNAedit.REDI"]
#all.patients_DF_RNA_small_subsonly[Database %in% c("yes_no"), Database := "RNAedit.RADAR"]

ggplot(all.patients_DF_RNA_small_subsonly, aes(x=Ref_Alt_coding, fill = RNAedit.REDI))+ #[!all.patients_DF_RNA_small_subsonly$Biotype_group ==
"Protein Coding",]
  geom_bar(stat = "count")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  labs(x="Nucleic acid changes Ref_Alt", y="Number of unique RNA alterations")+
  #scale_fill_manual(labels = c("RNA"), values = c("#619CFF"))+
  #facet_wrap(~ Biotype_group)+ #~
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    strip.text.x = element_text(size = 25, face = "bold"),
    axis.text.y = element_text(size = 25, vjust = 0.8),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")

###__ 7d #### :: Evaluate Ref_Alt of RNAonly mut (old)  _____________________________________________##########
```

```
## add Alt_Ref info
all.patients_DF_RNA_small$Ref_Alt <-  paste(all.patients_DF_RNA_small$REF, all.patients_DF_RNA_small$ALT, sep ="_")
all.patients_DF_RNA_small$Ref_Alt_short <- all.patients_DF_RNA_small$Ref_Alt

#collapse multi-subs
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^AA.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^CC.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^TT.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^GG.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^AC.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^AG.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^AT.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^CA.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^CG.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^CT.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^TA.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^TC.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^TG.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^GA.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^GC.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("^GT.*_", "multi_", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*AA$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*CC$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*TT$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*GG$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*CA$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*GA$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*TA$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*AC$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*GC$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*TC$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*AT$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*CT$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*GT$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*AG$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*CG$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)
all.patients_DF_RNA_small$Ref_Alt_short <- sub("_.*TG$", "_multi", all.patients_DF_RNA_small$Ref_Alt_short)

# plot Ref_Alt Info
ggplot(all.patients_DF_RNA_small, aes(x=Ref_Alt_short, fill = SOURCE))+
  geom_bar()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  labs(x="Amino acid changes Ref_Alt", y="Number of unique RNA alterations", fill = "Mutation origin")+
  scale_fill_manual(labels = c("RNA"), values = c("#619CFF"))+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 25, vjust = 0.8),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")

ggplot(all.patients_DF_RNA_small, aes(x=REF, y= ALT))+
  geom_point()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))

###__ 7e #### :: Check coverage & allele freq of RNAonly mut (old) ____________________________________##########
## calculate means of TumorVF and Coverage per unique Mutation
QC_RNAediting_VF <- all.patients_DF_RNA_small %>%
  group_by(Mutation_ID) %>%
  summarize(Gene = first(GENE), mean_TumorVF=mean(TumorVF), mean_Coverage=mean(Coverage),Sample_size=n(), Tumor_ID=Tumor_ID %>%
unique %>% sort %>% paste(collapse = " + "), EFFECT=EFFECT %>% unique %>% sort %>% paste(collapse = " + "), geneBiotype=geneBiotype %>%
unique %>% sort %>% paste(collapse = " + "))

## filter for VF over 5% and coverage over 10
QC_RNAediting_VF_filtered <- as.data.table(QC_RNAediting_VF)
QC_RNAediting_VF_filtered <- QC_RNAediting_VF_filtered[mean_TumorVF >= 0.05 & mean_Coverage >= 10]

# plot general distibution of VF and coverage
ggplot(all.patients_DF_RNA_small, aes(x = TumorVF)) +
  geom_histogram()+
  geom_point(aes(y= Coverage))+
  scale_y_continuous(labels = comma, limits = c(0,50000))+
  theme(text=element_text(size=20))
```

```
ggplot(all.patients_DF_RNA_small, aes(x = Coverage)) +
  geom_histogram()+
  scale_x_continuous(labels = comma, limits = c(0,500))

# plot means distribution of VF and coverage
ggplot(QC_RNAediting_VF, aes(x= mean_TumorVF, y=mean_Coverage))+
  geom_point()+
  scale_y_continuous(labels=comma, limits = c(0,400))
ggplot(QC_RNAediting_VF_filtered, aes(x= mean_TumorVF, y=mean_Coverage))+
  geom_point()+
  scale_y_continuous(labels=comma,limits = c(0,1000))
ggplot(QC_RNAediting_VF_filtered, aes(x = mean_TumorVF)) +
  geom_histogram()+
  geom_point(aes(y= mean_Coverage))+
  scale_y_continuous(labels = comma, limits = c(0,30000))+
  theme(text=element_text(size=20))

## for pre-filtered data from 10a: TumorVF and Coverage
QC_RNAediting_VF_prefiltered <- all.patients_DF_RNA_small_filtered %>%
  group_by(Mutation_ID) %>%
  summarize(Gene = first(GENE), mean_TumorVF=mean(TumorVF), mean_Coverage=mean(Coverage),Sample_size=n(), Tumor_ID=Tumor_ID %>%
unique %>% sort %>% paste(collapse = " + "), EFFECT=EFFECT %>% unique %>% sort %>% paste(collapse = " + "), geneBiotype=geneBiotype %>%
unique %>% sort %>% paste(collapse = " + "))

# plot general distribution of VF and coverage
ggplot(all.patients_DF_RNA_small_filtered, aes(x = TumorVF)) +
  geom_histogram()+
  geom_point(aes(y= Coverage))+
  scale_y_continuous(labels = comma, limits = c(0,50000))+
  theme(text=element_text(size=20))
ggplot(all.patients_DF_RNA_small_filtered, aes(TumorVF))+
  geom_density()
ggplot(all.patients_DF_RNA_small_filtered, aes(x = TumorVF)) +
  geom_histogram()

# plot means distribution
ggplot(QC_RNAediting_VF_prefiltered, aes(x= mean_TumorVF, y=mean_Coverage))+
  geom_point()+
  scale_y_continuous(labels=comma, limits = c(0,400))
ggplot(QC_RNAediting_VF_prefiltered, aes(x = mean_TumorVF)) +
  geom_histogram()+
  geom_point(aes(y= mean_Coverage))+
  scale_y_continuous(labels = comma, limits = c(0,30000))+
  theme(text=element_text(size=20))
```

6.9.2 Analysis whole immunopeptidome data

```
library(tidyverse)
library(openxlsx)
library(stringr)
library(reshape2)
library(ggplot2)
library(gplots)
library(ggrepel)
library(extrafont)
library(UpSetR)
library(RColorBrewer)
library(viridis)
library(data.table)
library(dplyr)
library(gridExtra)
library(pheatmap)
library(ComplexUpset)

source(file = "functions/import.references.R")
source("Peptides/Venn_diagrams_v2.R")

# PAC: alle Protein-ids, in denen das Peptid auftaucht
# MSMS Count: (?) Anzahl der Spectren
# gene_symbol: Name of gene (where peptide originates)
# gene: id of gene
# gene_biotype: Genetype of gene
# description:
# transcript:
# transcript_biotype:
```

```
# title: welche Rohdatei von Spectren
# charge:
# SQ: Sequence
# mod_sites:
# score:
# Spectra.Mass:
# Q-value:
# Theory.SQ.Mass:
# Delta.Mass:
# Delta.Mass.ppm:
# Specific.Flag:
# Label.Flag:
# Target_decoy:
# Chromosome:
# X.:
# Protein.AC:

#### (1) #### Import Raw Data from .txt-files ________________________________________________#####
path.files = "whole_peptidomics/raw_data/WT1FDR_Celina_annotated_filtered"
files.raw.list = as.vector(list.files(path = path.files, pattern = "*.tsv", full.names = T))
import_and_identify_filename <- function(x) {
  df <- read.delim2(file = x)
  name <- str_sub(x, start = 61, end = 75)  # adapt to file path length
  df["Master_ID"] <- name
  return(df)
}
data.raw2 <- lapply(files.raw.list, import_and_identify_filename)
data.raw2 <- plyr::rbind.fill(data.raw2)

#### (2) #### Clean and order ____________________________________________________________######
clean_and_order_new <- function(DF){
  temp <- DF %>%
    rename(Seq=Sq) %>%
    rename(MS.MS.count=MScount) %>%
    rename(PAC=Proteins) %>%
    rename(Spectra.Mass=Exp.MH.) %>%
    rename(Theory.Sq.Mass=Calc.MH.) %>%
    rename(Delta.Mass=Mass_Shift.Exp..Calc..) %>%
    rename(Mod_Sites=Modification) %>%
    rename(Target_Decoy=Target.Decoy) %>%
    rename(Label.Flag=Label) %>%
    select(Seq, Master_ID, gene_symbol, description, gene_biotype, transcript_biotype, everything(), -PAC, -gene_id, -transcript, -Title, PAC, gene_id, 
transcript, Title)
}
WP.2 <- clean_and_order_new(data.raw2)

#### (3) #### Adapt references and reference IN_# __________________________________________#######
WP.2$Master_ID_long <- as.factor(WP.2$Master_ID)
WP.2 <- WP.2 %>% separate(Master_ID, c("Patient_ID", "Master_ID_group", "Metastasis"), "_")
# adapt Mel15 entries to the ImmuNEO naming
WP.2$Metastasis[WP.2$Metastasis == "WTanno"] <- (WP.2$Master_ID_group[WP.2$Metastasis == "WTanno"]) # replace the Metastasis column 
entries,if WTanno, with the respective Mel15 metastasis name
WP.2$Master_ID_group[WP.2$Patient_ID == "Mel15"] <- (WP.2$Patient_ID[WP.2$Patient_ID == "Mel15"])
#create Master ID by combining two columns
WP.2$Master_ID = paste0(WP.2$Master_ID_group,"_",WP.2$Metastasis)
#combine with entity reference
WP.2 <- merge(WP.2, reference.entity) %>% #merges the reference.entity list
  select(Master_ID, Patient_ID, everything()) %>%  # sorts the df
  rename(gene_description="description")
WP.2$Patient_ID <- sub("NEO","IN_", WP.2$Patient_ID)
WP.2$gene_biotype <- sub("/IG_C_gene", "", WP.2$gene_biotype)
WP.2$gene_biotype <- sub("IG_C_gene/", "", WP.2$gene_biotype)
WP.2$gene_biotype <- sub("IG_V_gene/", "", WP.2$gene_biotype)
WP.2$gene_biotype <- sub("/polymorphic_pseudogene", "", WP.2$gene_biotype)

#### (4) Analysis 1: How many peptides to each Gene/Patient _________________________________######
## Collapse table into unqiue peptide per tumor
WP.2_dt <- as.data.table(WP.2)
wtpeps_counts <- WP.2_dt[, .N, by = Master_ID]
othercols <- c("Seq", "Master_ID")
mergecols <- setdiff(names(WP.2_dt), othercols)
WP.2_dt_unique <- WP.2_dt[, lapply(.SD, function(x){paste0(unique(x),collapse=";")}), .SDcols = mergecols, by=othercols]
#create Tumor ID by combining two columns
WP.2_dt_unique$Tumor_ID = paste0(WP.2_dt_unique$Patient_ID,"_",WP.2_dt_unique$Metastasis)
### DEBUG and check correct number of unique peptides --> all numbers the same?
wtpep_counts <- WP.2_dt_unique[, .N, by=.(Master_ID)]
```

```
wtpep_counts_2 <- WP.2_dt_unique[, .(number_of_distinct_peptides = uniqueN(Seq)), by = Master_ID]
wtpep_counts_3 <- WP.2_dt[, .(number_of_distinct_peptides = uniqueN(Seq)), by = Master_ID]
## Analysis for distinct data frames with specific information
# Number of unique peptides per patient and gene
WP.A2.1 <- WP.2[!WP.2$Patient_ID == "Mel15",] %>%
  distinct(Master_ID, Seq, gene_symbol, Metastasis, .keep_all = T) %>%
  group_by(gene_symbol, Master_ID) %>%
  summarise(N.unique.peptides.pergene.andMaster_ID=n(), gene_description=first(gene_description), Patient_ID=first(Patient_ID),
Tumor_entity=first(Tumor_entity), Metastasis =first(Metastasis)) %>%
  ungroup() %>%
  group_by(gene_symbol) %>%
  mutate(N.total.peptides.pergene=sum(N.unique.peptides.pergene.andMaster_ID))
# Number of unique peptides per gene
WP.A2.2 <- WP.2[!WP.2$Patient_ID == "Mel15",]  %>%
  distinct(Seq, gene_symbol, .keep_all = T) %>%
  group_by(gene_symbol) %>%
  summarise(N.unique.peptides.pergene=n(), gene_description=first(gene_description)) %>%
  select(-gene_description)

WP.Venn.Metastasis <- WP.2 %>%
  distinct(Master_ID, Seq, .keep_all = T)

WP.BT.group <- WP.2 %>%
  group_by(gene_biotype) %>%
  summarise(N.gene_biotype=n())

## Merge DFs and delete NA's and change format for ggplot
WP.A2 <- left_join(WP.A2.1, WP.A2.2, by=c("gene_symbol")) %>%
  filter(!is.na(gene_symbol))

WP.heat <- WP.A2 %>%
  select(-Patient_ID, -Tumor_entity, -gene_description) %>%
  pivot_wider(names_from = Master_ID, values_from = N.unique.peptides.pergene.andMaster_ID) %>%
  filter(!is.na(gene_symbol)) %>%
  arrange(desc(N.unique.peptides.pergene)) %>%
  ungroup() %>%
  top_n(100, N.unique.peptides.pergene)

WP.heat.2 <- as.matrix(WP.heat)
WP.heat.2[is.na(WP.heat.2)] <- 0
coul <- colorRampPalette(brewer.pal(8, "PiYG"))(25)
heat.top.100 <- heatmap(WP.heat.2, scale="column", col=coul)
legend(x="bottomright", legend=c("min", "ave", "max", "test","min", "ave", "max", "test"),
       fill=coul)

WP.heat[is.na(WP.heat)] <- 0
WP.heat <- WP.heat %>%
  pivot_longer(("1MULDR_T1":"Q1PB42_T3"), names_to="Master_ID", values_to =  "N.unique.peptides.pergene.andMaster_ID") %>%
  select(gene_symbol, Master_ID, N.unique.peptides.pergene.andMaster_ID, everything())

## Plot #1 --- Heatmap with ggplot ____________________________________________________________________#####
ggplot(data=WP.heat, aes(y=gene_symbol, x=Master_ID, fill=N.unique.peptides.pergene.andMaster_ID))+
  geom_tile(color='White', size=0.05)+
  scale_fill_viridis(name="# unique peptides per \n gene and Master_ID")+
  theme(axis.text.x = element_text(angle = 90))
cols <- hclust(dist(WP.heat))

## Plot #2 --- # of unique peptides per gene_________________________________________________________####
WP.A2.plot.2 <- WP.A2 %>%
  group_by(gene_symbol) %>%
  summarise(N.unique.peptides.pergene=first(N.unique.peptides.pergene), N.total.peptides.pergene=first(N.total.peptides.pergene)) %>%
  top_n(20, N.unique.peptides.pergene)

ggplot(WP.A2.plot.2[!WP.A2.plot.2$gene_symbol == "MAP4",], aes(x=reorder(gene_symbol, (-N.unique.peptides.pergene)),
y=N.unique.peptides.pergene))+
  geom_col()+
  labs(x="Gene", y="Number of unique peptides")+
  theme_PS()+
  theme(axis.text.x = element_text(size= 25),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25))+
  coord_flip()

### other analysis and plot for Top mutated genes
```

```
Top_pep_genes <- as.vector(unique(WP.A2.plot.2$gene_symbol))
Top_pep_genes <- Top_pep_genes[!Top_pep_genes %in% "MAP4"]
WP.2_topPepGenes <- subset(WP.2, gene_symbol %in% Top_pep_genes)

### filled by genes
nb.cols <- 20
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.cols)
ggplot(WP.2_topPepGenes[!WP.2_topPepGenes$Patient_ID == "Mel15",], aes(x=forcats::fct_infreq(Patient_ID), fill=gene_symbol))+
  geom_bar()+
  coord_flip()+
  labs(y="Number of unique peptides", x="Patient", fill="Gene")+
  theme_PS()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size= 25),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size= 25),
        legend.title = element_text(size=25, face = "bold"))+
  scale_fill_manual(values = mycolors)

### filled by Patients
nb.cols <- 32
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.cols)
ggplot(WP.2_topPepGenes[!WP.2_topPepGenes$Patient_ID == "Mel15",], aes(x=forcats::fct_infreq(gene_symbol), fill=Patient_ID))+
  geom_bar(colour = "black")+
  coord_flip()+
  labs(y="Number of unique peptides per Patient", x="Gene", fill="Patient ID")+
  theme_PS()+
  scale_y_continuous(breaks = c(100, 200, 300, 400, 500))+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size= 25),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size= 25),
        legend.title = element_text(size=25, face = "bold"))+
  guides(fill=guide_legend(ncol=8))+
  scale_fill_manual(values = mycolors)
#geom_text(data=all.patients_DF_uniquePep_topMutGenes[all.patients_DF_uniquePep_topMutGenes$Patient_ID ==
"IN_01",],aes(label=Biotype_group),stat="count", hjust=-6) # annotates Biotype_group

## Plot #3 A --- # of unique peptides per Master_ID ____________________________________________#####
WP.A2.plot.3 <- WP.A2.1 %>%
  group_by(Master_ID) %>%
  summarise(N.unique.peptides.perMaster_ID=sum(N.unique.peptides.pergene.andMaster_ID), Patient_ID=first(Patient_ID),
Metastasis=first(Metastasis), Tumor_entity=Tumor_entity %>% unique %>% sort %>% paste(collapse = ", ")) %>%
  ungroup()
WP.A2.plot.3.help <- WP.A2.plot.3 %>%
  group_by(Patient_ID) %>%
  mutate(Tumor_entity=Tumor_entity %>% unique %>% sort %>% paste(collapse = ", ")) %>%
  distinct(Patient_ID, .keep_all = T)
WP.A2.plot.3 <- WP.A2.plot.3 %>%
  select(-Tumor_entity) %>%
  left_join(WP.A2.plot.3.help) %>%
  group_by(Patient_ID) %>%
  mutate(max.N.unique.peptides.perMaster_ID=max(N.unique.peptides.perMaster_ID))

# Plot unique peptides per patient sorted by size
ggplot(data = WP.A2.plot.3, aes(x=reorder(Patient_ID, desc(max.N.unique.peptides.perMaster_ID)), y=N.unique.peptides.perMaster_ID,
fill=Metastasis))+
  geom_bar(position = "dodge", stat = "identity")+
  geom_text(aes(label=Tumor_entity, y=max.N.unique.peptides.perMaster_ID+500 ),nudge_x = -0.15, angle=35, size=5, color=c("#555555"),
fontface="plain", family = "sans", hjust=0)+
  labs( x="Patient ID", y="# of unique (in Seq) peptides")+
  scale_y_continuous(limits = c(0, 1.23*max(WP.A2.plot.3$N.unique.peptides.perMaster_ID)) )+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 25))+
  expand_limits(x=30)

# Plot unique peptides per patient unsorted
ggplot(data = WP.A2.plot.3, aes(x=Patient_ID, y=N.unique.peptides.perMaster_ID, fill=Metastasis))+
  geom_bar(position = "dodge", stat = "identity")+
  geom_text(aes(label=Tumor_entity, y=max.N.unique.peptides.perMaster_ID+500 ),nudge_x = -0.15, angle=90, size=5, color=c("#555555"),
fontface="plain", family = "sans", hjust=0)+
  labs( x="Patient ID", y="# of unique (in Seq) peptides")+
  scale_y_continuous(limits = c(0, 1.23*max(WP.A2.plot.3$N.unique.peptides.perMaster_ID)) )+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 25))+
  expand_limits(x=30)+
  scale_fill_brewer(palette = "Paired")

#Plot without Mel15
ggplot(data = WP.A2.plot.3[!WP.A2.plot.3$Patient_ID == "Mel15",], aes(x=Patient_ID, y=N.unique.peptides.perMaster_ID, fill=Metastasis))+
  geom_bar(position = "dodge", stat = "identity")+
  #geom_text(aes(label=Tumor_entity, y=max.N.unique.peptides.perMaster_ID+500 ),nudge_x = -0.15, angle=90, size=5, color=c("#555555"),
fontface="plain", family = "sans", hjust=0)+
  labs( x="Patient ID", y="Number of unique peptides")+
  #scale_y_continuous(limits = c(0, 1.23*max(WP.A2.plot.3$N.unique.peptides.perMaster_ID)) )+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25))+
  expand_limits(x=30)+
  scale_fill_brewer(palette = "Paired")

#Plot only core samples
WP.2_dt_unique_core <- WP.2_dt_unique[!Master_ID %in% c("GXL1B7_T2", "64EMZ9_T1", "Q1PB42_T1", "Q1PB42_T3",
"1MULDR_T1","1MULDR_T2","1MULDR_T3","NVDER5_T1", "LFNUX6_T2", "ATE46U_T1", "Mel15_T1", "Mel15_T2" )]
WP.2_dt_unique_core$Patient_ID <- sub("_", "-", WP.2_dt_unique_core$Patient_ID)
ggplot(WP.2_dt_unique_core, aes(x=Patient_ID))+
  geom_bar(position = "dodge", fill = "#1F78B4")+
  #geom_text(aes(label=Tumor_entity, y=max.N.unique.peptides.perMaster_ID+500 ),nudge_x = -0.15, angle=90, size=5, color=c("#555555"),
fontface="plain", family = "sans", hjust=0)+
  labs( x="Patient ID", y="Number of unique peptides")+
  #scale_y_continuous(limits = c(0, 1.23*max(WP.A2.plot.3$N.unique.peptides.perMaster_ID)) )+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25))+
  expand_limits(x=30)

## Plot #3 B --- # of unique peptides per Master_ID normal + quant ______________________________#####
WP.2.count.quant <- fread(file = "whole_peptidomics/export/Summary_peptidomics.csv")
WP.2.count.quant[WP.2.count.quant == "x"] <- NA

col_names <- colnames(WP.2.count.quant)
changeCols_2 <- col_names[! col_names %in% c("Tumor_ID","Patient_ID","Master_ID","Metastasis")]
WP.2.count.quant <- WP.2.count.quant[,(changeCols_2):= lapply(.SD, as.numeric), .SDcols = changeCols_2] # change to numeric values

ggplot(data = WP.2.count.quant, aes(x=Tumor_ID, y = quant_wt_peptides_1FDR))+ # wt_peptidome_1FDR, quant_wt_peptides_1FDR
  geom_bar(position = "dodge", stat = "identity", fill = "#A6CEE3")+ # "#1F78B4"-normal, "#A6CEE3" - quant
  labs( x="Patient ID", y="Number of unique peptides per gram tumor")+
  #scale_y_continuous(limits = c(0, 1.23*max(WP.A2.plot.3$N.unique.peptides.perMaster_ID)) )+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25))+
  expand_limits(x=30)

## for facet wrap
WP.2.count.quant.long <- fread(file = "whole_peptidomics/export/Summary_peptidomics_long.csv")
WP.2.count.quant.long[WP.2.count.quant.long == "x"] <- NA
```

```
col_names <- colnames(WP.2.count.quant.long)
changeCols_2 <- col_names[! col_names %in% c("Tumor_ID","Patient_ID","Master_ID","Metastasis", "Analysis")]
WP.2.count.quant.long <- WP.2.count.quant.long[,(changeCols_2):= lapply(.SD, as.numeric), .SDcols = changeCols_2] # change to numeric values
WP.2.count.quant.long.core <- WP.2.count.quant.long[!Master_ID %in% c("GXL1B7_T2", "64EMZ9_T1", "Q1PB42_T1", "Q1PB42_T3",
"1MULDR_T1","1MULDR_T2","1MULDR_T3","NVDER5_T1", "LFNUX6_T2", "ATE46U_T1" )]

ggplot(WP.2.count.quant.long.core, aes(x = Tumor_ID, y= Value, fill = Analysis)) +
  geom_bar(stat = "identity") +
  theme_PS()+
  facet_wrap(~factor(Analysis, levels=c("Total wt peptidome", "Quantified wt peptidome") ), ncol = 1, scales="free_y")+ #strip.position = "left"
  theme(
    axis.text.x = element_text(angle = 45, size = 20, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 20, face = "bold"),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")+
  scale_fill_brewer(palette = "Paired")+
  labs(x="Patient ID", y="Number of unique peptides")

## Plot #4 --- Venn Plot: Peptide overlap comparison for different Metastases _________________#######
for (i in unique((filter(WP.Venn.Metastasis, grepl("T2", Master_ID)|grepl("T4", Master_ID)))$Patient_ID)){
  POC <- WP.Venn.Metastasis %>%
    filter(Patient_ID==i) %>%
    distinct(Metastasis, Seq, .keep_all = T) %>%
    group_by(Metastasis) %>%
    summarise(peptide=paste0(Seq, collapse = ";"), N.peptides=n())
  T1 <- as.vector(str_split(POC$peptide[1],";", simplify = T))
  T2 <- as.vector(str_split(POC$peptide[2],";", simplify = T))
  T4 <- as.vector(str_split(POC$peptide[3],";", simplify = T))
  metastasis.found.in <- list("T1"=T1, "T2"=T2, "T4"=T4)
  metastasis.found.in <- metastasis.found.in[!is.na(metastasis.found.in)]

  Venn.plot.2.wt(metastasis.found.in, save.plot = T, paste0("WP.Venn_", i))
}




## Plot #5 --- General plots of wt peptide data _________________________________________________#####
ggplot(WP.2, aes(x = gene_biotype))+
  geom_bar(stat = 'count')+
  scale_y_log10()

#### (5) Analysis 2: Peptide overlapps UPSET PLOT ______________________________________________#############
## for peptides
plot.DF.1 <- WP.2[!WP.2$Patient_ID == "Mel15",] %>%
  mutate(Seq_temp=Seq) %>%
  select(Seq, Seq_temp, gene_symbol, gene_biotype, MS.MS.count, Master_ID, Patient_ID, Metastasis) %>%
  mutate(Tumor_ID=paste(Patient_ID,Metastasis, sep = "_")) %>%
  mutate_if(is.factor, as.character) %>%
  distinct(Tumor_ID, Seq_temp, .keep_all = T) %>%
  group_by(Tumor_ID) %>%
  mutate(grouped_id = row_number()) %>%
  spread(Tumor_ID, Seq_temp) %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  mutate_at(c(6:ncol(.)), ~replace(., .!=0, 1)) %>%
  as.data.frame() %>%
  mutate_at(vars(c(6:ncol(.))), as.numeric) %>%
  #select(-Id, -gene_symbol, -gene_biotype, -grouped_id) %>%
  group_by(Seq) %>%
  summarise_all(list(~ max(.))) %>% # depricated: summarise_all(funs(max))
  as.data.frame()

## for Genes
plot.DF.2 <- WP.2[!WP.2$Patient_ID == "Mel15",] %>%
  mutate(Gene_temp=gene_symbol) %>%
  select(Gene_temp, gene_symbol, gene_biotype, MS.MS.count, Master_ID, Patient_ID, Metastasis) %>%
  mutate(Tumor_ID=paste(Patient_ID,Metastasis, sep = "_")) %>%
  mutate_if(is.factor, as.character) %>%
  distinct(Tumor_ID, Gene_temp, .keep_all = T) %>%
  group_by(Tumor_ID) %>%
  mutate(grouped_id = row_number()) %>%
  spread(Tumor_ID, Gene_temp) %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  mutate_at(c(6:ncol(.)), ~replace(., .!=0, 1)) %>%
  as.data.frame() %>%
  mutate_at(vars(c(6:ncol(.))), as.numeric) %>%
  #select(-Id, -gene_symbol, -gene_biotype, -grouped_id) %>%
  group_by(gene_symbol) %>%
  summarise_all(list(~ max(.))) %>% # depricated: summarise_all(funs(max))
  as.data.frame()

###### (5a) for all peptides _______________________________________________________________________#####
### 5.1 create plots
plot_upset <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), group.by = "sets", cutoff = 5, order.by = "freq")}

plot_upset_2 <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), group.by = "sets", cutoff = 5, order.by = "freq", nintersects =  NA)}
UpSet_plot <-plot_upset_2(plot.DF.1)
UpSet_plot <-plot_upset_2(plot.DF.2)

plot_upset_3 <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), order.by = "freq", decreasing = T, nintersects = 80)}
UpSet_plot <-plot_upset_3(plot.DF.1)
UpSet_plot <-plot_upset_3(plot.DF.2)

UpSet_plot

Patients <- colnames(plot.DF.1)[-1:-8]
upset(plot.DF.1, Patients, name = 'Seq', width_ratio = 0.1,n_intersections=30,
      base_annotations=list('Intersection size'=intersection_size(counts=FALSE,mapping=aes(fill=gene_symbol))))

upset(plot.DF.1, Patients,name = 'Seq', base_annotations=list('Intersection size'=intersection_size(counts=FALSE)+ theme(legend.position = "none")), width_ratio = 0.2,height_ratio = 1.7, n_intersections =100,sort_sets=FALSE, themes=upset_default_themes(text=element_text(size=20)))

### 5.2 extract peptide overlapps in list
# get list of columns
# for peptides
RawMatrix <- plot.DF.1[, -1:-8] # only get data columns
RawList <- list() # init list
for(col in colnames(RawMatrix)){
  seqcol <- "Seq"
  plot.DF.1 <- data.table(plot.DF.1)
  tmp <- plot.DF.1[, .(get(seqcol), get(col))]
  seqs <- tmp[V2 == 1, V1]
  RawList[[ col ]] <- seqs # append vector to list using name from col
}

#for genes
RawListGenes <- list() # init list
for(col in colnames(RawMatrix)){
  seqcol <- "gene_symbol"
  plot.DF.1 <- data.table(plot.DF.1)
  tmp <- plot.DF.1[, .(get(seqcol), get(col))]
  seqs <- tmp[V2 == 1, V1]
  RawListGenes[[ col ]] <- seqs # append vector to list using name from col
}

# source of this function: https://github.com/hms-dbmi/UpSetR/issues/85#issuecomment-327900647
fromList <- function (input) {
  elements <- unique(unlist(input))
  data <- unlist(lapply(input, function(x) {
    x <- as.vector(match(elements, x))
  }))
  data[is.na(data)] <- as.integer(0)
  data[data != 0] <- as.integer(1)
  data <- data.frame(matrix(data, ncol = length(input), byrow = F))
  data <- data[which(rowSums(data) != 0), ]
  names(data) <- names(input)
  row.names(data) <- elements
  return(data)
}

overlapGroups <- function (listInput, sort = TRUE) {
  listInputmat    <- fromList(listInput) == 1
  listInputunique <- unique(listInputmat)
  grouplist <- list()
  for (i in 1:nrow(listInputunique)) {
    currentRow <- listInputunique[i,]
    myelements <- which(apply(listInputmat,1,function(x) all(x == currentRow)))
    attr(myelements, "groups") <- currentRow
    grouplist[[paste(colnames(listInputunique)[currentRow], collapse = ":")]] <- myelements
    myelements
  }
  if (sort) {
    grouplist <- grouplist[order(sapply(grouplist, function(x) length(x)), decreasing = TRUE)]
  }
  attr(grouplist, "elements") <- unique(unlist(listInput))
  return(grouplist)
}

li.pep <- overlapGroups(RawList)
saveRDS(li, "Upset_Matrix_LIgroups.rds")
Upset_Matrix_LIgroups <- readRDS("Upset_Matrix_LIgroups.rds")

li.pep.gene <- overlapGroups(RawListGenes)
li.pep.gene[["IN_01_T1:IN_02_T1:IN_03_T1:IN_04_T1:IN_05_T1:IN_08_T1:IN_09_T1:IN_11_T1:IN_11_T2:IN_13_T1:IN_14_T1:IN_15_T1:IN_16_T1:IN_17_T1:IN_17_T2:IN_17_T3:IN_18_T1:IN_19_T1:IN_19_T2:IN_19_T3:IN_19_T4:IN_20_T1:IN_22_T1:IN_23_T1:IN_23_T2:IN_24_T1:IN_24_T2:IN_25_T1:IN_26_T1:IN_27_T1:IN_27_T2:IN_28_T1:IN_30_T1:IN_31_T1:IN_32_T1:IN_33_T1:IN_34_T1:IN_35_T1:IN_36_T1:IN_37_T1:IN_38_T1"]]

###### (5b) for tumor-associated peptides ________________________________________________________#####
############## 5.1 filter data set for tumor-asoociated proteins ___________________________________#####
# load reference gene/protein list from ProteinAtlas https://www.proteinatlas.org/humanproteome/tissue/cancer on 06.04.2021
reference.tumorProteins <- fread(file = "rawfiles/references/protein_class_COSMIC.tsv")
reference.tumorProteins <- reference.tumorProteins$Gene

plot.DF.1.filtered <- subset(plot.DF.1, gene_symbol %in% reference.tumorProteins)
plot.DF.2.filtered <- subset(plot.DF.2, gene_symbol %in% reference.tumorProteins)
colnames(plot.DF.1.filtered) <- gsub("IN_", "IN-", colnames(plot.DF.1.filtered))
colnames(plot.DF.1.filtered) <- gsub("_T", "-T", colnames(plot.DF.1.filtered))

############## 5.2 extract overview on shared peptides in list __________________________________#####
# create column with all samples where mutation is present
plot.DF.1.filtered <- as.data.table(plot.DF.1.filtered)
dt <- plot.DF.1.filtered[, -2:-8]
patnames <- gsub(" ", "",as.data.table(melt(dt, "Seq"))[,toString(variable[value==1]), Seq]$V1)
plot.DF.1.filtered[, Samples := patnames]
setcolorder(plot.DF.1.filtered, c(colnames(plot.DF.1.filtered)[1:8], 'Samples'))

# expand matrix and add perPatient information
dt <- plot.DF.1.filtered[, -2:-8]
meltDT <- as.data.table(melt(dt, "Seq"))
meltDT[, variable := gsub("_T\\d$", "", variable)]
meltDT <- meltDT[, max(value), by=.(Seq, variable)]
library(tidyr)
dtnew <- spread(meltDT, variable, V1)
patnames <- gsub(" ", "",as.data.table(melt(dtnew, "Seq"))[,toString(variable[value==1]), Seq]$V1)
dtnew[, Patients := patnames]
setcolorder(dtnew, c('Seq', 'Patients'))

plot.DF.1.filtered.summary <- merge(plot.DF.1.filtered[,1:9],dtnew[,1:2])
plot.DF.1.filtered.summary <- plot.DF.1.filtered.summary[,-5:-8]

# count number of samples/patients
sample_count <- lengths(regmatches(plot.DF.1.filtered.summary$Samples, gregexpr(",", plot.DF.1.filtered.summary$Samples)))+1
plot.DF.1.filtered.summary[, count_samples := sample_count]
patient_count <- lengths(regmatches(plot.DF.1.filtered.summary$Patients, gregexpr(",", plot.DF.1.filtered.summary$Patients)))+1
plot.DF.1.filtered.summary[, count_patients := patient_count]
#filtering step if wanted
#plot.DF.1.filtered.summary <- plot.DF.1.filtered.summary[count_patients >= 10]
plot.DF.1.filtered.summary.counted <- plot.DF.1.filtered.summary[,.N, by = count_patients]

############## 5.3 Bar plot sharing patients overview _____________________________________________######
# for unfiltered matrix
ggplot(plot.DF.1.filtered.summary[plot.DF.1.filtered.summary$count_patients >=4], aes(x= factor(count_patients)))+
  geom_bar(stat = "count", position = "stack")+
  #scale_x_discrete(drop = FALSE,limits =c("4", "5", "6", "7", "8","9", "10", "11", "12", "13", "14"))+
  #scale_y_continuous(breaks=seq(0, 20, 2))+
  labs(x="Number of sharing patients", y="Number of unique MS tumor-associated peptides")+
  theme_PS()+
  theme(#legend.position = "bottom",
    axis.text.x = element_text(size= 20),
    axis.text.y = element_text(size= 20),
    plot.title = element_blank(),
    axis.title.y =  element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 25, face = "bold"),
    legend.text = element_text(size= 20),
    legend.title = element_text(size=25, face = "bold"),
    legend.position = "bottom")+
  guides(fill=guide_legend(ncol=2))

############# 5.3 Upset plots tumor sample overlap  _____________________________________________######
# final overview plot --> save as device size 15x20
upset(plot.DF.1.filtered, Patients,name = 'Seq', base_annotations=list('Intersection size'=intersection_size(counts=FALSE)+ theme(legend.position = "none")), width_ratio = 0.2,height_ratio = 1.7, n_intersections =100,sort_sets=FALSE, themes=upset_default_themes(text=element_text(size=20)))

############# 5.4 extract peptide overlappsper Upset plot in list _____________________________######
Upset_Matrix <- fread("/home/rad/Downloads/Upset_Matrix_filtered.csv")

# get list of columns
RawMatrix <- plot.DF.1.filtered[, -1:-8] # only get data columns
RawList <- list() # init list
for(col in colnames(RawMatrix)){
  seqcol <- "Seq"
  plot.DF.1.filtered <- data.table(plot.DF.1.filtered)
  tmp <- plot.DF.1.filtered[, .(get(seqcol), get(col))]
  seqs <- tmp[V2 == 1, V1]
  RawList[[ col ]] <- seqs # append vector to list using name from col
}

# for genes
RawListGenes <- list() # init list
for(col in colnames(RawMatrix)){
  seqcol <- "gene_symbol"
  plot.DF.1.filtered <- data.table(plot.DF.1.filtered)
  tmp <- plot.DF.1.filtered[, .(get(seqcol), get(col))]
  seqs <- tmp[V2 == 1, V1]
  RawListGenes[[ col ]] <- seqs # append vector to list using name from col
}

# source of this function: https://github.com/hms-dbmi/UpSetR/issues/85#issuecomment-327900647
fromList <- function (input) {
  elements <- unique(unlist(input))
  data <- unlist(lapply(input, function(x) {
    x <- as.vector(match(elements, x))
  }))
  data[is.na(data)] <- as.integer(0)
  data[data != 0] <- as.integer(1)
  data <- data.frame(matrix(data, ncol = length(input), byrow = F))
  data <- data[which(rowSums(data) != 0), ]
  names(data) <- names(input)
  row.names(data) <- elements
  return(data)
}
overlapGroups <- function (listInput, sort = TRUE) {
  listInputmat    <- fromList(listInput) == 1
  listInputunique <- unique(listInputmat)
  grouplist <- list()
  for (i in 1:nrow(listInputunique)) {
    currentRow <- listInputunique[i,]
    myelements <- which(apply(listInputmat,1,function(x) all(x == currentRow)))
    attr(myelements, "groups") <- currentRow
    grouplist[[paste(colnames(listInputunique)[currentRow], collapse = ":")]] <- myelements
    myelements
  }
  if (sort) {
    grouplist <- grouplist[order(sapply(grouplist, function(x) length(x)), decreasing = TRUE)]
  }
  attr(grouplist, "elements") <- unique(unlist(listInput))
  return(grouplist)
  # save element list to facilitate access using an index in case rownames are not named
}

li.pep.filtered <- overlapGroups(RawList)
saveRDS(li, "Upset_Matrix_filtered_LIgroups.rds")
# shared 8 peptides 1
```

```
li.pep.filtered[["IN_02_T1:IN_05_T1:IN_08_T1:IN_09_T1:IN_11_T1:IN_11_T2:IN_13_T1:IN_14_T1:IN_15_T1:IN_16_T1:IN_18_T1:IN_23_T1:IN_24_T1:IN_24_T2:IN_25_T1:IN_32_T1:IN_36_T1:IN_37_T1:IN_38_T1"]]
# shared 8 peptides 2
li.pep.filtered[["IN_05_T1:IN_08_T1:IN_11_T1:IN_11_T2:IN_14_T1:IN_15_T1:IN_16_T1:IN_18_T1:IN_23_T1:IN_25_T1:IN_36_T1:IN_37_T1"]]
# shared 2 peptides
li.pep.filtered[["IN_02_T1:IN_05_T1:IN_08_T1:IN_09_T1:IN_11_T1:IN_11_T2:IN_13_T1:IN_14_T1:IN_15_T1:IN_16_T1:IN_18_T1:IN_23_T1:IN_24_T1:IN_24_T2:IN_25_T1:IN_32_T1:IN_35_T1:IN_36_T1:IN_37_T1:IN_38_T1"]]

sum(WP.2$Seq == "CA")

li.pe.filtered.gene <- overlapGroups(RawListGenes)
saveRDS(li, "Upset_Matrix_filtered_genes_LIgroups.rds")

Upset_Matrix_filtered_genes_LIgroups <- readRDS("whole_peptidomics/export/Upset_Matrix_filtered_genes_LIgroups.rds")

###### (5c) for Top20 genes with peptides ____________________________________________________________#####
### 5.1 filter data set for Top20 genes
plot.DF.1.Top20 <- subset(plot.DF.1, gene_symbol %in% Top_pep_genes)
plot.DF.2.Top20 <- subset(plot.DF.2, gene_symbol %in% Top_pep_genes)
plot_upset_2 <- function(data){
  upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), group.by = "sets", cutoff = 5, order.by = "freq", show.numbers = FALSE , nintersects =  40)
}

#exclude subsets with less then 10 genes (37 patients * 5 subsets = 185 intersects)
plot_upset_4 <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), group.by = "sets", cutoff = 5, order.by = "freq", nintersects =  185)}

plot_upset_4 <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), group.by = "sets", cutoff = 3, order.by = "freq", nintersects =  100)}
UpSet_plot_filtered <-plot_upset_4(plot.DF.1.Top20)
UpSet_plot_filtered <-plot_upset_4(plot.DF.2.Top20)
UpSet_plot_filtered

plot_upset_3 <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), order.by = "freq", nintersects =  150)}
UpSet_plot_filtered <-plot_upset_3(plot.DF.1.Top20)
UpSet_plot_filtered <-plot_upset_3(plot.DF.2.Top20)
UpSet_plot_filtered

## with ComplexUpset
upset(plot.DF.2.Top20, Patients, base_annotations=list('Intersection size'=intersection_size(counts=FALSE,mapping=aes(fill=gene_symbol))+ theme(legend.position = "none")), name = 'Filtered for tumor associated genes', width_ratio = 0.1,height_ratio = 0.5, n_intersections=14)

###### (5d) for CTA peptides _______________________________________________________________________#####
############# 5.1 filter data set for CTA proteins ______________________________________________######
#load reference gene list from CTDatabase http://www.cta.lncc.br/index.php?id=4 on 10.03.2021
CTA_ref_table <- read.csv(file = "rawfiles/references/CTAs_CTDatabase.csv")
CTA_ref <- CTA_ref_table$Gene_symbol

############# 5.2 Upset Plot ________________________________________________________________________######
#Matrix
plot.DF.1.CTA <- subset(plot.DF.1, gene_symbol %in% CTA_ref)
plot.DF.1.CTA <- plot.DF.1[plot.DF.1$gene_symbol == "PRAME",]
plot.DF.2.CTA <- subset(plot.DF.2, gene_symbol %in% CTA_ref)

### Upset plot variant 1
plot_upset_2 <- function(data){
  upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), group.by = "sets", cutoff = 5, order.by = "freq", show.numbers = FALSE , nintersects =  40)
}

#exclude subsets with less then 10 genes (37 patients * 5 subsets = 185 intersects)
plot_upset_4 <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), group.by = "sets", cutoff = 5, order.by = "freq", nintersects =  185)}

plot_upset_4 <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), group.by = "sets", cutoff = 3, order.by = "freq", nintersects =  100)}
UpSet_plot_filtered <-plot_upset_4(plot.DF.1.CTA)
UpSet_plot_filtered <-plot_upset_4(plot.DF.2.CTA)
UpSet_plot_filtered

plot_upset_3 <- function(data){upset(data, sets =  c(colnames(data)[grep("IN_01", colnames(plot.DF.1)):ncol(data)]), mb.ratio = c(0.5, 0.5), order.by = "freq", nintersects =  150)}
UpSet_plot_filtered <-plot_upset_3(plot.DF.1.CTA)
UpSet_plot_filtered <-plot_upset_3(plot.DF.2.CTA)
```

UpSet_plot_filtered

```
### Upset plot variant 2
Patients <- colnames(plot.DF.1.CTA)[-1:-8]
upset(plot.DF.1.CTA, Patients, base_annotations=list('Intersection size'=intersection_size(counts=FALSE,mapping=aes(fill=gene_symbol))+
theme(legend.position = "none")),width_ratio = 0.1,height_ratio = 1, n_intersections=150)

upset(plot.DF.1.CTA, Patients,mode = 'exclusive_intersection',sort_sets=FALSE,min_degree = 3, base_annotations=list('Intersection
size'=intersection_size(counts=TRUE,mapping=aes(fill=gene_symbol))+ theme(legend.position =
"none")+guides(fill=guide_legend(ncol=15))),themes=upset_default_themes(text=element_text(size=20)), width_ratio = 0.1,height_ratio = 1)
#mode = 'inclusive_intersection' / 'exclusive_intersection'
#themes=upset_default_themes(text=element_text(size=20))

############# 5.3 extract peptide overlapps from Upset plot in list ____________________________#####
# get list of columns
RawMatrix <- plot.DF.1.CTA[, -1:-8] # only get data columns
RawList <- list() # init list
for(col in colnames(RawMatrix)){
  seqcol <- "Seq"
  plot.DF.1.CTA <- data.table(plot.DF.1.CTA)
  tmp <- plot.DF.1.CTA[, .(get(seqcol), get(col))]
  seqs <- tmp[V2 == 1, V1]
  RawList[[ col ]] <- seqs # append vector to list using name from col
}

# for genes
RawListGenes <- list() # init list
for(col in colnames(RawMatrix)){
  seqcol <- "gene_symbol"
  plot.DF.1.CTA <- data.table(plot.DF.1.CTA)
  tmp <- plot.DF.1.CTA[, .(get(seqcol), get(col))]
  seqs <- tmp[V2 == 1, V1]
  RawListGenes[[ col ]] <- seqs # append vector to list using name from col
}

# source of this function: https://github.com/hms-dbmi/UpSetR/issues/85#issuecomment-327900647
fromList <- function (input) {
  elements <- unique(unlist(input))
  data <- unlist(lapply(input, function(x) {
    x <- as.vector(match(elements, x))
  }))
  data[is.na(data)] <- as.integer(0)
  data[data != 0] <- as.integer(1)
  data <- data.frame(matrix(data, ncol = length(input), byrow = F))
  data <- data[which(rowSums(data) != 0), ]
  names(data) <- names(input)
  row.names(data) <- elements
  return(data)
}
overlapGroups <- function (listInput, sort = TRUE) {
  listInputmat    <- fromList(listInput) == 1
  listInputunique <- unique(listInputmat)
  grouplist <- list()
  for (i in 1:nrow(listInputunique)) {
    currentRow <- listInputunique[i,]
    myelements <- which(apply(listInputmat,1,function(x) all(x == currentRow)))
    attr(myelements, "groups") <- currentRow
    grouplist[[paste(colnames(listInputunique)[currentRow], collapse = ":")]] <- myelements
    myelements
  }
  if (sort) {
    grouplist <- grouplist[order(sapply(grouplist, function(x) length(x)), decreasing = TRUE)]
  }
  attr(grouplist, "elements") <- unique(unlist(listInput))
  return(grouplist)
}

li.pep.CTA <- overlapGroups(RawList)
#saveRDS(li, "Upset_Matrix_filtered_LIgroups.rds")

# shared 8 peptides 1
li.pep.filtered[["IN_02_T1:IN_05_T1:IN_08_T1:IN_09_T1:IN_11_T1:IN_11_T2:IN_13_T1:IN_14_T1:IN_15_T1:IN_16_T1:IN_18_T1:IN_23_T1:IN_24_T1
:IN_24_T2:IN_25_T1:IN_32_T1:IN_36_T1:IN_37_T1:IN_38_T1"]]
# shared 8 peptides 2
li.pep.filtered[["IN_05_T1:IN_08_T1:IN_11_T1:IN_11_T2:IN_14_T1:IN_15_T1:IN_16_T1:IN_18_T1:IN_23_T1:IN_25_T1:IN_36_T1:IN_37_T1"]]
# shared 2 peptides
```

```
li.pep.filtered[["IN_02_T1:IN_05_T1:IN_08_T1:IN_09_T1:IN_11_T1:IN_11_T2:IN_13_T1:IN_14_T1:IN_15_T1:IN_16_T1:IN_18_T1:IN_23_T1:IN_24_T1
:IN_24_T2:IN_25_T1:IN_32_T1:IN_35_T1:IN_36_T1:IN_37_T1:IN_38_T1"]]

sum(WP.2$Seq == "CA")

li.pep.CTA.gene <- overlapGroups(RawListGenes)
saveRDS(li, "Upset_Matrix_filtered_genes_LIgroups.rds")

############# 5.4 extract count overview per patient/sample _____________________________________######
# create column with all samples where mutation is present = perSample info
plot.DF.1.CTA <- as.data.table(plot.DF.1.CTA)
dt <- plot.DF.1.CTA[, -2:-8]
patnames <- gsub(" ", "",as.data.table(melt(dt, "Seq"))[,toString(variable[value==1]), Seq]$V1)
plot.DF.1.CTA[, Samples := patnames]
setcolorder(plot.DF.1.CTA, c(colnames(plot.DF.1.CTA)[1:5], 'Samples'))

# expand matrix and add perPatient information
dt <- plot.DF.1.CTA[, -2:-9]
meltDT <- as.data.table(melt(dt, "Seq"))
meltDT[, variable := gsub("_T\\d$", "", variable)]
meltDT <- meltDT[, max(value), by=.(Seq, variable)]
library(tidyr)
dtnew <- spread(meltDT, variable, V1)
patnames <- gsub(" ", "",as.data.table(melt(dtnew, "Seq"))[,toString(variable[value==1]), Seq]$V1)
dtnew[, Patients := patnames]
setcolorder(dtnew, c('Seq', 'Patients'))
plot.DF.1.CTA_summary <- merge(plot.DF.1.CTA[,1:6],dtnew[,1:2])

# count number of samples/patients
sample_count <- lengths(regmatches(plot.DF.1.CTA_summary$Samples, gregexpr(",", plot.DF.1.CTA_summary$Samples)))+1
plot.DF.1.CTA_summary[, count_samples := sample_count]
patient_count <- lengths(regmatches(plot.DF.1.CTA_summary$Patients, gregexpr(",", plot.DF.1.CTA_summary$Patients)))+1
plot.DF.1.CTA_summary[, count_patients := patient_count]
plot.DF.1.CTA_summary_filtered <- plot.DF.1.CTA_summary[count_patients >= 3]
peptide_counts_CTA <- plot.DF.1.CTA_summary[,.N, by = patient_count]

ggplot(plot.DF.1.CTA_summary, aes(x= factor(count_patients)))+
  geom_bar(stat = "count", position = "stack")+
  scale_y_continuous(breaks = seq(0, 160, by = 20))+
  labs(x="Number of sharing patients", y="Number of unique CTA peptides")+
  theme_PS()+
  theme(#legend.position = "bottom",
    axis.text.x = element_text(size= 20),
    axis.text.y = element_text(size= 20),
    plot.title = element_blank(),
    axis.title.y =  element_text(size=25),
    axis.title.x = element_text(size=25),
    strip.text = element_text(size = 25, face = "bold"),
    legend.text = element_text(size= 20),
    legend.title = element_text(size=25, face = "bold"),
    legend.position = "bottom")+
  guides(fill=guide_legend(ncol=2))

############# 5.5 Ggplot plot overview  shared CTA peptides _____________________________________######
# Data table
WP.2.CTA <- WP.2[!WP.2$Patient_ID == "Mel15",]%>%
  mutate(Tumor_ID=paste(Patient_ID,Metastasis, sep = "_"))

WP.2.CTA <- WP.2.CTA[WP.2.CTA$gene_symbol %in% CTA_ref,]
WP.2.CTA <- as.data.table(WP.2.CTA)
peptide_counts_CTA <- WP.2.CTA[,.N, by = Seq]
WP.2.CTA_counts <- merge(WP.2.CTA, peptide_counts_CTA, by = "Seq")
WP.2.CTA_counts_filteres <- WP.2.CTA_counts[WP.2.CTA_counts$N >= 2]

nb.cols <- 29
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.cols)
ggplot(WP.2.CTA_counts[WP.2.CTA_counts$N >= 3], aes(x= Seq, fill = Patient_ID)) +
  geom_bar(colour = "black") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(x ="CTA gene \n Peptide sequence", y = "Number of unique CTA peptides", fill = "Patient ID") +
  #facet_wrap(~Group, scales = "free")+
  theme(legend.key.size = unit(1, "cm"),
      axis.text.x = element_text(size= 20),
      axis.text.y = element_text(size= 20),
      axis.title.y = element_text(size=25),
      axis.title.x = element_text(size=25),
      strip.text.x = element_text(size = 18, angle = 90),
      #strip.placement = "outside",                   # Place facet labels outside x axis labels.
      #strip.background = element_rect(fill = "white"),  # Make facet label background white.
      legend.text = element_text(size= 20),
      legend.title = element_text(size=25, face = "bold")) +
  scale_y_continuous(breaks=c(0,2,4,6,8,10,12,14))  +
  #scale_fill_manual(values=sample(col_vector, n))+
  scale_fill_manual(values = mycolors) +
  #scale_fill_brewer(palette = "Set3") +
  facet_grid(~gene_symbol,
          scales = "free_x", # Let the x axis vary across facets.
          space = "free_x",  # Let the width of facets vary and force all bars to have the same width.
          switch = "x")

#### (6) Analysis 3: Peptide length Distribution ______________________________________________###############
WP.pep.length <- WP.2 %>%
  distinct(Master_ID, Seq, chromosome, .keep_all = T) %>%
  mutate(pep.length=str_length(Seq)) %>%
  mutate(Tumor_ID=paste(Patient_ID,Metastasis, sep = "_")) %>%
  arrange(Master_ID, pep.length) %>%
  group_by(Master_ID, Patient_ID, Tumor_ID, pep.length) %>%
  summarise(N=n()) %>%
  arrange(Master_ID, pep.length)
  #filter(Patient_ID=="IN_19")
WP.pep.length$pep.length <- factor(WP.pep.length$pep.length)
WP.pep.length$Tumor_ID <- sub("_", "-", WP.pep.length$Tumor_ID)

ggplot(WP.pep.length, aes(x=Tumor_ID, y=N, fill=pep.length))+
  geom_col(position = "dodge")+
  #geom_text(aes(label=Patient_ID, y=(-1000)), nudge_x = -0.4, angle=45, size=5, color=c("#555555"), fontface="plain", family = "sans", hjust=0)+
  theme_PS()+
  scale_y_continuous(breaks = seq(0,20000,4000))+
  theme(axis.text.x = element_text(angle = 90))+
  labs(x = "Master ID", y="N", fill="Peptide length")

ggplot(WP.pep.length[! WP.pep.length$Patient_ID == "Mel15",], aes(x=pep.length, y=N, fill=pep.length))+
  geom_col(position = "dodge")+
  theme_PS()+
  theme(axis.text.x = element_blank(),
      axis.title.x = element_blank(),
      panel.grid.minor = element_blank(),
      panel.grid.major = element_blank(),
      strip.text.x = element_text(size = 13))+
  labs(y="Number of peptides", fill="Peptide \nlength [AA]")+
  facet_wrap(~Tumor_ID, scales = "free") +
  scale_fill_brewer(palette = "Paired")
  #scale_fill_hue(l=65, c=100)

#### (7) Analysis 4: Export data for MHC-motif deconvolution___________________________________________________________########
WP.Gibbs <- WP.2 %>%
  group_by(Patient_ID) %>%
  #mutate(Seq_ID=letters[row_number()]) %>%
  mutate(Seq_ID=sprintf("%05d", row_number())) %>%
  mutate(Seq_ID=paste0(Patient_ID,sep="_",Seq_ID,sep="_",Seq)) %>%
  select(Seq_ID,everything()) %>%
  mutate(Seq_ID=paste0(sep=">",Seq_ID)) %>%
  mutate(Seq=as.character(Seq)) %>%
  ungroup() %>%
  select(Seq,Patient_ID) %>%
  as.data.frame() #%>%
  #filter(Patient_ID=="IN_09")

export.peptide.list <- function(DF) {temp <- DF %>%
    select(Seq)
  write_delim(temp, path = paste0("whole_peptidomics/Gibbs_Clustering/FASTAs/WP.Gibbs.", DF$Patient_ID[1], ".csv"), delim = "", col_names = F)
}

for (i in unique(WP.Gibbs$Patient_ID)){WP.Gibbs.selected <- WP.Gibbs %>%
    filter(Patient_ID==i)
  export.peptide.list(WP.Gibbs.selected)
}

#### (8) Analysis 5: CTA analysis heatmap _____________________________________________________#####
#load reference gene list from CTDatabase http://www.cta.lncc.br/index.php?id=4 on 10.03.2021
CTA_ref_table <- read.csv(file = "rawfiles/references/CTAs_CTDatabase.csv")
```

```
CTA_ref <- CTA_ref_table$Gene_symbol

# generate matrix for heatmap with rownames etc
WP.A2$Tumor_ID = paste0(WP.A2$Patient_ID,"_",WP.A2$Metastasis)

WP.heat_CTA_INonly <- WP.A2 %>%
  select(-Patient_ID, -Tumor_entity, -gene_description, -Metastasis,- Master_ID, -N.total.peptides.pergene, -N.unique.peptides.pergene) %>%
  pivot_wider(names_from = Tumor_ID, values_from = N.unique.peptides.pergene.andMaster_ID) %>%
  filter(!is.na(gene_symbol)) %>%
  ungroup()
WP.heat_CTA_INonly[is.na(WP.heat_CTA_INonly)] <- 0
WP.heat_CTA_INonly <- subset(WP.heat_CTA_INonly, gene_symbol %in% CTA_ref)

rownames_heat <- WP.heat_CTA_INonly$gene_symbol
WP.heat_CTA_INonly$gene_symbol <- NULL
row.names(WP.heat_CTA_INonly) <- rownames_heat

WP.heat_CTA_INonly <- as.matrix(WP.heat_CTA_INonly)
WP.heat_CTA_INonly <- WP.heat_CTA_INonly[, sort(colnames(WP.heat_CTA_INonly))]

# plot heatmap
annotate_entity_CTA <- annotate_entity_all[,.(Tumor_entity_short, Tumor_ID)]
annotate_entity_CTA <- as.data.frame(annotate_entity_CTA)
rownames_anno_CTA <- annotate_entity_CTA$Tumor_ID
row.names(annotate_entity_CTA) <- rownames_anno_CTA
annotate_entity_CTA$Tumor_ID <- NULL
colnames(annotate_entity_CTA) <- "Entity"

#Change Naming of patients
colnames(WP.heat_CTA_INonly) <- gsub("_", "-", colnames(WP.heat_CTA_INonly))
rownames(annotate_entity_CTA) <- gsub("_", "-", rownames(annotate_entity_CTA))

pheatmap(WP.heat_CTA_INonly,cluster_rows = FALSE ,cluster_cols = FALSE, fontsize=12, annotation_col = annotate_entity_CTA, color =
heat.colors(100, rev = TRUE))

# add count info to sort genes in decreasing order
WP.heat_CTA_INonly.count <- cbind(WP.heat_CTA_INonly, Total= as.numeric(rowSums(WP.heat_CTA_INonly)))
WP.heat_CTA_INonly.count <- WP.heat_CTA_INonly.count[order(WP.heat_CTA_INonly.count[, "Total"], decreasing = TRUE),]

pheatmap(WP.heat_CTA_INonly.count[,1:41],cluster_rows = FALSE ,cluster_cols = FALSE, fontsize=12, annotation_col = annotate_entity_CTA, color
= heat.colors(100, rev = TRUE))
```

6.9.3 Analysis predicted neoantigen candidates

```
library(data.table)
library(ggplot2)
library(scales)
library(ggpubr)
library(readr)
library(xlsx)
library(tidyverse)
library(splitstackshape)
library(Vennerable)
library(openxlsx)
library(ggsci)

# ________ Analyze prediction data set ______________________________________________________________________________________#####
##_______________ set working directory and load data
_____________________________________________________________________######
IN_pep_prediciton_all <- fread("netmhcResults_allPatients.tsv")
IN_pep_prediciton_all
#IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <- fread("IN_pep_prediction_allPatients_filtered_uniquePep_uniqueHLA.csv")

##_______________ filtering the data set for pre-defined criteria _______________________________________________________######
IN_pep_prediction_all_filtered <- IN_pep_prediciton_all[bindlevel == "SB" | bindlevel == "WB" ]
IN_pep_prediction_all_filtered <- IN_pep_prediction_all_filtered[affinity <= 200 ]

##_______________ collapsing the data to unique peptides ________________________________________________________________#####
# Step 1: collapse all columns of unique peptides for same patient and same HLA
mergecols <- c("affinity", "affScore","rankP","bindlevel","chr","pos","ref","alt","pep17", "pep17marked","calledBy")
othercols <- setdiff(names(IN_pep_prediction_all_filtered), mergecols)
IN_pep_prediction_all_filtered_uniquePep <- IN_pep_prediction_all_filtered[, lapply(.SD, function(x){paste0(unique(x),collapse=";")}),
                                        .SDcols = mergecols, by=othercols]
```

```
# Step 2: collapse all columns of unique peptides (combine HLAs)
mergecols <- c("HLA", "affinity", "affScore","rankP","bindlevel","chr","pos","ref","alt","pep17", "pep17marked","calledBy")
othercols <- setdiff(names(IN_pep_prediction_all_filtered_uniquePep), mergecols)
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <- IN_pep_prediction_all_filtered_uniquePep[, lapply(.SD,
function(x){paste0(unique(x),collapse=";")}),.SDcols = mergecols, by=othercols]
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$bindlevel <- gsub("WB;SB", "SB;WB",
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$bindlevel ) #rename multiple bindlevels the same way
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$calledBy <-gsub("Mutect2,Mutect2", "Mutect2",
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$calledBy )
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$calledBy <-gsub("StrelkaRNA,StrelkaRNA", "StrelkaRNA",
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$calledBy )
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <- IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[grep(",", calledBy),
calledBy := "Mutect2 + StrelkaRNA"]
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <- IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[grep(";", calledBy),
calledBy := "Mutect2 + StrelkaRNA"]

IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <- IN_pep_prediction_all_filtered_uniquePep_uniqueHLA %>% mutate( patient.2 =
patient)
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <- cSplit(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, "patient.2", "_")
colnames(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA) <- gsub("patient.2_1", "Master_ID",
colnames(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA))
colnames(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA) <- gsub("patient.2_2", "Metastasis",
colnames(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA))
colnames(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA) <- gsub("patient", "Tumor_ID",
colnames(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA))

IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <- merge(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, reference.master,
by = "Master_ID")
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Tumor_ID <-
paste(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Patient_ID,
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Metastasis, sep ="_")
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Mutation_ID <- paste(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$chr,
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$pos,IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$ref,
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$alt, sep ="_")

setorder(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, Tumor_ID)
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <-
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[!IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Tumor_ID == "IN_25_T2",]

##_______________ print data as text file split by patient ____________________________________________________________#####
#list for every tumor
mydataset <- IN_pep_prediction_all_filtered_uniquePep_uniqueHLA
changethisnamehere <- "Prediction_netMHC_peplist_"

for(ipatient in unique(mydataset$patient)){
  print(ipatient)
  out <- mydataset[patient == ipatient]
  outfilename <- paste0(changethisnamehere, ipatient, ".tsv")
  fwrite(out, outfilename, col.names = T, sep = "\t")
}

# full data set
write_csv(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, path =
"Filtered_lists/IN_pep_prediction_allPatients_filtered_uniquePep_uniqueHLA.csv" )
write.xlsx(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, file
="Filtered_lists/IN_pep_prediction_allPatients_filtered_uniquePep_uniqueHLA.xlsx" )

# counts
predicted_peptides_counts <- IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[, .N, by=.(Tumor_ID)]
predicted_peptides_counts_Calledby <- IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[, .N, by=.(Tumor_ID, calledBy)]

##_______________ plots _______________________________________________________________________________#######
### ___________________ plots for quick data evaluation _______________________________________________________#######
ggplot(Remaining_SB_1, aes(x=affinity, fill = HLA)) +
	geom_bar()

ggplot(Remaining_SB_2, aes(x=affinity, fill = HLA)) +
  geom_bar()

ggplot(IN_pep_prediction_all_affinityfiltered, aes(patient))+
  geom_bar()+
  geom_text(stat = "count", aes(label = ..count..), vjust = -1)

### ___________________ plot variant 1 - unique peptides per tumor + counts annotated ________________________________#####
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, aes(x=Tumor_ID)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(title="Number of unique predicted peptides filtered by bind level only SB per patient tumor",
       x ="Patient tumor ID", y = "Number of unique peptides") +
  scale_y_continuous(breaks=pretty_breaks(n = 10))+
  geom_text(stat = "count", aes(label = ..count..), vjust = -1)

### ___________________ plot variant 1b - unique peptides per tumor + counts annotated + mutational load added _____#####
Mutation_data_perTumor <- as.data.table(mutational_load.permaster)
Mutation_data_perTumor$Tumor_ID <- paste(Mutation_data_perTumor$Patient_ID, Mutation_data_perTumor$Metastasis, sep = "_")

ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, aes(x=Tumor_ID)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(title="Number of unique predicted peptides filtered by affinity and bindlevel per patient tumor",
       x ="Patient tumor ID", y = "Number of unique peptides") +
  scale_y_continuous(breaks=pretty_breaks(n = 10))+
  geom_text(stat = "count", aes(label = ..count..), vjust = -1)+
  geom_line(data = Mutation_data_perTumor, aes(y = mut_load/10 ), colour = "red", group = 1)+
  scale_y_continuous(sec.axis = sec_axis(~./10, name = "Mutationl load *10"))

### ___________________ plot variant 2 - unique peptides per patient + bind level info _______________________________########
# without Mel15
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[!IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Patient_ID ==
"Mel15",], aes(x= Patient_ID, fill = bindlevel)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(title="Number of unique predicted peptides filtered by affinity and bind level per patient",
       x ="Patient ID", y = "Number of unique peptides") +
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 15),
        plot.title = element_text(size= 20),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 15),
        legend.title = element_text(size=20, face = "bold")) +
  scale_y_continuous(breaks=pretty_breaks(n = 10))

## ordered
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, aes(x= forcats::fct_infreq(Patient_ID), fill = bindlevel)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(title="Number of unique predicted peptides filtered by affinity and bind level per patient",
       x ="Patient ID", y = "Number of unique peptides") +
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 15),
        plot.title = element_text(size= 20),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 15),
        legend.title = element_text(size=20, face = "bold")) +
  scale_y_continuous(breaks=pretty_breaks(n = 10))+
  scale_x_discrete(breaks=labelorder,labels=labelname)

### ___________________ plot variant 3 - unique peptides per tumor + bindlevel _______________________________________######
## only ImmuNEOs / without Mel15
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[!IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Patient_ID ==
"Mel15",], aes(x= Tumor_ID, fill = bindlevel)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.5, "cm"),
        axis.text.x = element_text(size = 20),
      axis.text.y = element_text(size = 20),
      plot.title = element_blank(),
      axis.title.y = element_text(size=25),
      axis.title.x = element_text(size=25),
      legend.position = "bottom",
      legend.text = element_text(size = 20),
      legend.title = element_text(size=25)) +
 labs(x ="Patient Tumor ID", y = "Number of predicted 9mer peptides", fill = "Bindlevel") +
 scale_y_continuous(breaks=pretty_breaks(n = 10))+
 scale_fill_hue(l=65, c=70)+
 scale_fill_discrete(labels = c("SB", "SB & WB", "WB"))

## only Mel15
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Patient_ID ==
"Mel15",], aes(x= Tumor_ID, fill = bindlevel)) +
 geom_bar() +
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1, size = 20),
      axis.title.y = element_blank(),
      axis.text.y = element_text(size = 20),
      plot.title = element_blank(),
      legend.text = element_text(size = 20),
      legend.title = element_text(size=25),
      axis.title.x = element_text(size=25)) +
 #theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
 labs(x ="", y = "Number of unique peptides") +
 scale_y_continuous(breaks=pretty_breaks(n = 10))+
 scale_fill_hue(l=65, c=70)

### ___________________ plot variant 4 - unique peptides per tumor + calledBy _________________________________________######
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, aes(x= Tumor_ID, fill = calledBy)) +
 geom_bar() +
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
 theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
 labs(title="Number of unique predicted peptides filtered by bind level only SB per patient tumor",
     x ="Patient tumor ID", y = "Number of unique peptides") +
 scale_y_continuous(breaks=pretty_breaks(n = 10))

ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA_V1, aes(x= forcats::fct_infreq(Tumor_ID), fill = bindlevel)) +
 geom_bar() +
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
 theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
 labs(title="Number of unique predicted peptides filtered by affinity and bind level per patient tumor",
     x ="Patient tumor ID", y = "Number of unique peptides") +
 scale_y_continuous(breaks=pretty_breaks(n = 10))

### ___________________ plot variant 5 - unique peptides + HLA plot and combine multiple HLA ________________________#######
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA_V1 <- IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[grep(";", HLA), HLA :=
"multipleHLA"]

## only ImmuNEOs / without Mel15
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA_V1[!IN_pep_prediction_all_filtered_uniquePep_uniqueHLA_V1$Patient_ID
== "Mel15",], aes(x= Tumor_ID, fill = HLA)) +
 geom_bar() +
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
 theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 10), legend.title = element_text(size = 10)) +
 labs(title="Number of unique predicted peptides per patient tumor",
     x ="Patient tumor ID", y = "Number of unique peptides") +
 scale_y_continuous(breaks=pretty_breaks(n = 10)) +
 guides(fill=guide_legend(ncol=1))
 #ylim(0,1250)+
 #scale_fill_manual(values = c("multipleHLA" = "grey"))

## only Mel15
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA_V1[IN_pep_prediction_all_filtered_uniquePep_uniqueHLA_V1$Patient_ID
== "Mel15",], aes(x= Tumor_ID, fill = HLA)) +
 geom_bar() +
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
 theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 10), legend.title = element_text(size = 10)) +
 labs(title="Number of unique predicted peptides per patient tumor",
     x ="Patient tumor ID", y = "Number of unique peptides") +
 scale_y_continuous(breaks=pretty_breaks(n = 10)) +
 guides(fill=guide_legend(ncol=1))
```

```
#ylim(0,1250)+
#scale_fill_manual(values = c("multipleHLA" = "grey"))

### ___________________ plot variant 6 - unique peptides plotted by HLA-type __________________________________________#####
ggplot(IN_pep_prediction_all_filtered_uniquePep, aes(x= forcats::fct_infreq(HLA), fill = bindlevel)) +
 geom_bar() +
 coord_flip()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
 theme(legend.key.size = unit(0.2, "cm"),
       axis.text.x = element_text(size = 20),
       axis.text.y = element_text(size = 15),
       plot.title = element_blank(),
       axis.title.y = element_text(size=25),
       axis.title.x = element_text(size=25),
       legend.text = element_text(size = 20),
       legend.title = element_text(size=25)) +
 labs(x ="HLA type", y = "Number of unique peptides") +
 scale_y_continuous(breaks=pretty_breaks(n = 10))
 #scale_fill_discrete(name = "bindlevel", labels = c("none", "SB", "SB & WB", "WB"))
 #geom_text(stat = "count", aes(label = ..count..), vjust = 1)

### ___________________ plot variant 7 - unique peptides affinity distribution _______________________________________#####
ggplot(IN_pep_prediction_all_filtered_uniquePep_uniqueHLA, aes(x = affinity)) +
 geom_bar( stat = "count")

# _________ compare prediction data set with MS data set ______________________________________________________________######
#load already finished data if applicable
In_pep_combined_unique <- fread("UniquePeptides_bothPipeline_combined_V2.csv")

## ___________ load the two data sets ________________________________________________________________________________#####
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA <- fread("IN_pep_prediction_allPatients_filtered_uniquePep_uniqueHLA_??.csv")

IN_pep_MS_all <- fread("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/27 R scripts Philipp und
Niklas/Philipp/Peptides/rawfiles/Peptides_2021/IN.peptides.filtered.final.all.csv")
IN_pep_MS_all <- as.data.table(DF.plot) #if already loaded from "Plot_peptide_data_4_CSW.R" script


## ___________ adapt and sub-set the data tables to later merge them _____________________________________________####
colnames(IN_pep_MS_all) <- gsub("Seq", "peptide", colnames(IN_pep_MS_all))
colnames(IN_pep_MS_all) <- gsub("Master_ID_group", "Master_ID", colnames(IN_pep_MS_all))
IN_pep_MS_all <- IN_pep_MS_all %>% unite(Tumor_ID, c(Patient_ID, Metastasis), sep = "_", remove = FALSE)
IN_pep_MS_all_V2 <- IN_pep_MS_all[, c(1:33)]
IN_pep_MS_all_V2 <- IN_pep_MS_all_V2[,-c(6:32)]
IN_pep_MS_all_V2 <- IN_pep_MS_all_V2[mutationType == "substitution"]
IN_pep_MS_all_V2 <- IN_pep_MS_all_V2[, pipeline := "MS-based"]
#IN_pep_MS_all_V2 <- IN_pep_MS_all_V2[1:80]

IN_pep_prediciton_all_V2 <- IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[, c(1:3, 16,17)]
IN_pep_prediciton_all_V2 <- IN_pep_prediciton_all_V2[, pipeline := "prediction"]

## ___________ merge both data tables ________________________________________________________________________________####
In_pep_combined <- rbind(IN_pep_MS_all_V2, IN_pep_prediciton_all_V2, fill = TRUE)

mergecols <- c("pipeline","Master_ID","Patient_ID","Metastasis")
othercols <- c("peptide", "Tumor_ID")
In_pep_combined_unique <- In_pep_combined[, lapply(.SD, function(x){paste0(unique(x),collapse=";")}),
                                          .SDcols = mergecols, by=othercols]
In_pep_combined_unique[ ,Patient_ID := gsub(";NA", "", Patient_ID)]
In_pep_combined_unique[ ,Tumor_ID := gsub(";NA", "", Tumor_ID)]
In_pep_combined_unique[ ,Metastasis := gsub(";NA", "", Metastasis)]
In_pep_combined_unique <- In_pep_combined_unique[!In_pep_combined_unique$Master_ID == "Mel15",]

## ___________ tidy the data and add ImmuNEO_IDs _____________________________________________________________________####
In_pep_combined_unique <- In_pep_combined_unique %>% mutate( patient.2 = patient)
In_pep_combined_unique[, Master_ID_group := NULL]
In_pep_combined_unique[, Metastasis := NULL]
In_pep_combined_unique <- cSplit(In_pep_combined_unique, "patient.2", "_")
colnames(In_pep_combined_unique) <- gsub("patient.2_1", "Master_ID", colnames(In_pep_combined_unique))
colnames(In_pep_combined_unique) <- gsub("patient.2_2", "Metastasis", colnames(In_pep_combined_unique))
IN_IDs <- fread("Z:/AG-Krackhardt/ImmuNEO project/1 ImmuNEO-F?lle/Table_PatientIDs_core_cohort.csv")
IN_IDs <- fread("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/1 ImmuNEO-Fälle/Table_PatientIDs_core_cohort.csv")
labelorder <- IN_IDs[[1]]
```

```
labelname <- IN_IDs[[2]]
setorder(IN_IDs, MasterID)

## ____________ get the raw counts for each pipeline overall ____________________________________________________________#####
neoantigen_counts_allPipelines <- In_pep_combined_unique[, .N, by=.( pipeline)]

## ____________ get the raw counts for each pipeline per patient ________________________________________________________#####
neoantigen_counts_allPipelines_perPatient <- In_pep_combined_unique[, .N, by=.(Patient_ID, pipeline)]
setorder(neoantigen_counts_allPipelines_perPatient, Patient_ID)
neoantigen_counts_allPipelines_perPatient <- neoantigen_counts_allPipelines_perPatient %>% mutate( Patient = Patient_ID)

## ____________ get the raw counts for each pipeline per patient tumor _________________________________________________#####
neoantigen_counts_allPipelines_perTumor <- In_pep_combined_unique[, .N, by=.(patient, pipeline)]
setorder(neoantigen_counts_allPipelines_perTumor, patient)
write_csv(neoantigen_counts_allPipelines_perTumor, path = "neoantigen_counts_allPipelines_perTumor_V2.csv" )
write.xlsx(neoantigen_counts_allPipelines_perTumor, file ="neoantigen_counts_allPipelines_perTumor_V2.xlsx" )

## ____________ Filter for only 9mers from MS data set ___________________________________________________________________####
In_pep_combined_unique[, length := nchar(In_pep_combined_unique$peptide)] # count the characters of the peptide sequence and add as new column
In_pep_combined_unique_9mers <- In_pep_combined_unique[length == 9]
In_pep_combined_unique_9mers <- data.table(In_pep_combined_unique_9mers)
neoantigen_counts_allPipelines_9mers <- In_pep_combined_unique_9mers[, .N, by=.( pipeline)]

### ______________ plot the data _______________________________________________________________________________________#####
#### ________________ Bar charts _______________________________________________________________________________________#####
ggplot(In_pep_combined_unique_9mers, aes(x=Master_ID, fill = pipeline)) +
 geom_bar() +
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
 theme(legend.key.size = unit(0.2, "cm"),
    axis.text.x = element_text(size= 15),
    axis.text.y = element_text(size= 15),
    plot.title = element_text(size= 20),
    axis.title.y = element_text(size=20),
    axis.title.x = element_text(size=20),
    legend.text = element_text(size= 15),
    legend.title = element_text(size=20, face = "bold"),
    legend.position="bottom") +
 labs(title="Number of unique peptides identified by both pipelines per patient",
   x ="Patient ID", y = "Number of unique peptides") +
 scale_y_sqrt() +
 scale_x_discrete(limits=labelorder,
          labels=labelname)

# _________ Analyse peptides from shared mutations ______________________________________________________________________######
# Ref Table for shared mutations
shared_mutations_ref <- fread("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/27 R scripts Philipp und Niklas/Philipp/mutation_calling/Results_export/Shared_mutations_RNAall_summary_V2new.csv")
shared_mutations_ref <- shared_mutations_ref[,!2:5]

## see mutational analysis script
shared_mutations <- shared_mutations_DNA
shared_mutations <- shared_mutations_RNA
shared_mutations <- shared_mutations_RNA_Upset
shared_mutations_ref <- shared_mutations_DNA_ref
shared_mutations_ref <- shared_mutations_RNA_ref
shared_mutations_ref <- shared_mutations_RNA_Upset_ref

#shared mutations all
shared_mutations <- shared_mutations_ref$Mutation_ID
#shared mutations RNAall Group 1
shared_mutations <- c("1_45694928_T_C","14_50440012_C_T", "14_52775636_T_C", "15_41163832_T_C", "15_41299144_A_G", "15_75353745_A_G", "15_84641599_T_C", "16_81030835_A_G", "19_13773270_A_G", "19_13773604_A_G", "21_33264079_A_G")
#shared mutations RNAall Group 4
shared_mutations <- c("20_4629698_C_T","20_4629706_G_A", "20_4629727_C_T")
#shared mutations RNAall Group 8
shared_mutations <- c("20_57488521_C_A","20_57488540_T_C")
#shared mutations RNAall Group 12
shared_mutations <- c("10_50093288_G_A","10_50113947_G_C")

## ___________ whole data set __________________________________________________________________________________________########
# Step 1: filter data for INonly and shared mutations
```

```
Prediciton_pep_shared_mutations_all <- IN_pep_prediciton_all[!IN_pep_prediciton_all$patient == "Mel15_T1",]
Prediciton_pep_shared_mutations_all <- Prediciton_pep_shared_mutations_all[!Prediciton_pep_shared_mutations_all$patient ==
"Mel15_T2",]
Prediciton_pep_shared_mutations_all$Mutation_ID <- paste(Prediciton_pep_shared_mutations_all$chr,
Prediciton_pep_shared_mutations_all$pos,Prediciton_pep_shared_mutations_all$ref, Prediciton_pep_shared_mutations_all$alt, sep
="_")
Prediciton_pep_shared_mutations_all <- Prediciton_pep_shared_mutations_all[Prediciton_pep_shared_mutations_all$Mutation_ID %in%
shared_mutations,]
Prediciton_pep_shared_mutations_all$bindlevel <- sub("^$", "NB", Prediciton_pep_shared_mutations_all$bindlevel)

# Step 2: collapse all columns of unique peptides for same patient and same HLA
mergecols <- c("HLA", "affinity", "affScore","rankP","bindlevel","chr","pos","ref","alt","pep17", "pep17marked","calledBy", "Mutation_ID")
othercols <- setdiff(names(Prediciton_pep_shared_mutations_all), mergecols)
Prediciton_pep_shared_mutations_all <- Prediciton_pep_shared_mutations_all[, lapply(.SD,
function(x){paste0(unique(x),collapse=";")}),.SDcols = mergecols, by=othercols]

# Step 3: Tidy data
Prediciton_pep_shared_mutations_all$bindlevel <- sub(";NB", "", Prediciton_pep_shared_mutations_all$bindlevel)
Prediciton_pep_shared_mutations_all$bindlevel <- sub("NB;", "", Prediciton_pep_shared_mutations_all$bindlevel)
Prediciton_pep_shared_mutations_all$bindlevel <- sub("WB;SB", "SB;WB", Prediciton_pep_shared_mutations_all$bindlevel)

# Step 4: Add gene reference
Prediciton_pep_shared_mutations_all <- merge(Prediciton_pep_shared_mutations_all, shared_mutations_ref, by = "Mutation_ID")

# Step 5: Plot data
ggplot(Prediciton_pep_shared_mutations_all, aes(x= Gene, fill = bindlevel)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  labs(x ="Patient ID", y = "Number of unique peptides") +
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 15),
        plot.title = element_text(size= 20),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 15),
        legend.title = element_text(size=20, face = "bold"))

ggplot(Prediciton_pep_shared_mutations_all, aes(x= bindlevel, fill = Gene)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  #scale_y_log10()+
  labs(x ="Bind Level", y = "Number of unique peptides") +
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 15),
        plot.title = element_text(size= 20),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 15),
        legend.title = element_text(size=20, face = "bold"))+
  scale_fill_brewer(palette ="Paired")

## ___________ filtered data set __________________________________________________________________________________________######
Prediciton_pep_shared_mutations <-
IN_pep_prediction_all_filtered_uniquePep_uniqueHLA[!IN_pep_prediction_all_filtered_uniquePep_uniqueHLA$Patient_ID == "Mel15",]
Prediciton_pep_shared_mutations <- Prediciton_pep_shared_mutations[Prediciton_pep_shared_mutations$Mutation_ID %in%
shared_mutations,]
Prediciton_pep_shared_mutations_DNA <- merge(Prediciton_pep_shared_mutations, shared_mutations_ref, by = "Mutation_ID")
Prediciton_pep_shared_mutations_RNA <- merge(Prediciton_pep_shared_mutations, shared_mutations_ref, by = "Mutation_ID")
Prediciton_pep_shared_mutations_RNA_UpSet <- merge(Prediciton_pep_shared_mutations, shared_mutations_ref, by = "Mutation_ID")

peptide_counts <- Prediciton_pep_shared_mutations_DNA[,.N, by = peptide]
Prediciton_pep_shared_mutations_DNA <- merge(Prediciton_pep_shared_mutations_DNA, peptide_counts, by = "peptide")
Prediciton_pep_shared_mutations_DNA <- fread("Filtered_lists/20211122_Predcted_peptides_shared_SomaticMut_min4.csv") ## read
table with peptide sequence marked

peptide_counts <- Prediciton_pep_shared_mutations_RNA[,.N, by = peptide]
Prediciton_pep_shared_mutations_RNA <- merge(Prediciton_pep_shared_mutations_RNA, peptide_counts, by = "peptide")
peptide_counts <- Prediciton_pep_shared_mutations_RNA_UpSet[,.N, by = peptide]
Prediciton_pep_shared_mutations_RNA_UpSet <- merge(Prediciton_pep_shared_mutations_RNA_UpSet, peptide_counts, by = "peptide")
```

```
### _____________ plot the data _______________________________________________________________________#####
ggplot(Prediciton_pep_shared_mutations, aes(x= Tumor_ID, fill = GENE)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(x ="Patient ID", y = "Number of unique peptides") +
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 15),
        plot.title = element_text(size= 20),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 15),
        legend.title = element_text(size=20, face = "bold")) +
  scale_y_continuous(breaks=pretty_breaks(n = 10))
#scale_x_discrete(limits=labelorder, labels = labelname)
#geom_text(stat = "count", aes(label = ..count..), vjust = -1)

ggplot(Prediciton_pep_shared_mutations_DNA[Prediciton_pep_shared_mutations_DNA$N >= 4], aes(x= Patient_ID, fill =GENE)) +
  geom_bar(colour = "black") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(x ="Patient ID", y = "Number of unique peptides", fill = "Gene") +
  #facet_wrap(~Group, scales = "free")+
  theme(legend.key.size = unit(1, "cm"),
        axis.text.x = element_text(size= 20),
        axis.text.y = element_text(size= 20),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size= 20),
        legend.title = element_text(size=25, face = "bold")) +
  scale_y_continuous(breaks=pretty_breaks(n = 10))  +
  #scale_fill_brewer(palette = "Set3")
  guides(fill = guide_legend(ncol = 1))

## by sequence, annotate Tumor ID and Gene --> check for peptides that are predicted in many patients --> filter by number of patients having the peptide
### for somatic mutations
nb.cols <- 17
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.cols)
ggplot(Prediciton_pep_shared_mutations_DNA, aes(x= peptide, fill = Tumor_ID)) +
  geom_bar(colour = "black") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(x ="Gene \n Peptide sequence (mutated amino acid marked)", y = "Number of predicted 9mer peptides", fill = "Tumor ID") +
  #facet_wrap(~Group, scales = "free")+
  theme(legend.key.size = unit(1, "cm"),
        axis.text.x = element_text(size= 20),
        axis.text.y = element_text(size= 20),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        strip.text.x = element_text(size = 18, angle = 90),
        #strip.placement = "outside",                    # Place facet labels outside x axis labels.
        #strip.background = element_rect(fill = "white"),  # Make facet label background white.
        legend.text = element_text(size= 20),
        legend.title = element_text(size=25, face = "bold")) +
  scale_y_continuous(breaks=pretty_breaks(n = 10))  +
  #scale_fill_manual(values=sample(col_vector, n))+
  scale_fill_manual(values = mycolors) +
  #scale_fill_brewer(palette = "Set3") +
  facet_grid(~GENE,
             scales = "free_x", # Let the x axis vary across facets.
             space = "free_x",  # Let the width of facets vary and force all bars to have the same width.
             switch = "x")

#### for RNA alterations shared - Upset groups : "Prediciton_pep_shared_mutations_RNA_UpSet[Prediciton_pep_shared_mutations_RNA_UpSet$N >= 10]" or "Prediciton_pep_shared_mutations_RNA_UpSet_filtered"
nb.cols <- 28
mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(nb.cols)
ggplot(Prediciton_pep_shared_mutations_RNA_UpSet_filtered, aes(x= peptide, fill = Tumor_ID)) +
  geom_bar(colour = "black") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(legend.key.size = unit(0.2, "cm"), legend.text = element_text(size = 7), legend.title = element_text(size = 7)) +
  labs(x ="Gene \n Peptide sequence (mutated amino acid marked)", y = "Number of predicted 9mer peptides", fill = "Tumor ID") +
  #facet_wrap(~Group, scales = "free")+
  theme(legend.key.size = unit(1, "cm"),
        axis.text.x = element_text(size= 18),
        axis.text.y = element_text(size= 20),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25),
        strip.text.x = element_text(size = 18, angle = 90),
        #strip.placement = "outside",                   # Place facet labels outside x axis labels.
        #strip.background = element_rect(fill = "white"),  # Make facet label background white.
        legend.text = element_text(size= 20),
        legend.title = element_text(size=25, face = "bold")) +
  scale_y_continuous(breaks=pretty_breaks(n = 10))  +
  #scale_fill_manual(values=sample(col_vector, n))+
  scale_fill_manual(values = mycolors) +
  #scale_fill_brewer(palette = "Set3") +
  facet_grid(~Gene,
             scales = "free_x", # Let the x axis vary across facets.
             space = "free_x",  # Let the width of facets vary and force all bars to have the same width.
             switch = "x")
```

6.9.4 Analysis MS-based neoantigen candidates

```
library(tidyverse)
library(ggplot2)
library(RColorBrewer)
library(data.table)

#source(file = "ImmuNeo_peptides_all_V6.R")
if(!exists("Venn.plot.2", mode="function")) source("Venn_diagrams_v2.R")
#source(file = "functions/import.references.R")

#### Themes ____________________________________________________________________________________________________####
theme_PS <- function(){
  theme(
    plot.title=element_text(size=20, hjust = 0.5),
    plot.background = element_rect(fill = "transparent",colour = NA),
    panel.grid.major = element_line(color = "grey", linetype = "dotted", size=0.8),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "transparent",colour = NA),
    panel.border = element_rect(color = "white", fill = NA),
    #axis.line = element_line(color = "grey"),
    axis.ticks = element_line(color = "grey"),
    axis.text = element_text(size = 16),
    axis.text.x = element_text(angle = 0),
    axis.title = element_text(size = 18,face="bold"),
    legend.text = element_text(size = 18),
    legend.title = element_text(size= 18,face="bold")
  )
}

#### Load Data __________________________________________________________________________________________________####
DF.plot <- fread("rawfiles/Peptides_2021_10_ErrorSamples/IN.peptides.filtered.final.all_new_V2.csv")
DF.plot <- DF.plot[1:91]
DF.plot <- as.data.frame(DF.plot)

# subset for reactive neoantigens
#add reactive neoantigen from HD data
reactive_neoantigens <- append(reactive_neoantigens, "IN_19_c")
DF.plot.reactive <- subset(DF.plot, Seq_ID_short %in% reactive_neoantigens)

### 1 ## Candidate assessments general ____________________________________________________________________________#####
## 1a ## Peptide length distribution  _____________________________________________________________________________####
ggplot(DF.plot, aes(x=Peptide_length, fill=transcriptTypes))+
  geom_bar()+
  theme_PS()+
  theme(axis.text.x = element_text(size = 20),
      axis.text.y = element_text(size = 20),
      plot.title = element_blank(),
      axis.title.y = element_text(size=25),
      axis.title.x = element_text(size=25),
      legend.text = element_text(size = 20),
      legend.title = element_text(size=25),
      legend.position = "bottom")+
  labs(x= "Peptide length [AA]", y="Number of MS neoantigen candidates", fill="Transcript Type")+
  scale_x_continuous(breaks = c(8,9,10,11,12,13,14,15))+
  scale_fill_brewer(palette = "Paired")
```

```
### 2 ## Peptides per Patient distribution __________________________________________________________________________________________#####
##### 2a ## PLOT PD: Peptide distribution -- per Patient_ID ______________________________________________________________________#####
DF.plot.2 <- DF.plot %>%
  group_by(Patient_ID) %>%
  mutate(N.peptides=n()) %>%
  ungroup()

ggplot(data = DF.plot.2, aes(Patient_ID, fill=transcriptTypes))+
  geom_histogram(stat = "count", alpha=0.8)+
  labs(title = "Peptide distribution", x="Patient ID", y="# peptides")+
  scale_y_continuous(breaks = seq(0,22,2))+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 35))

ggplot(data = DF.plot.2, aes(reorder(Patient_ID,desc(N.peptides)), fill=transcriptTypes))+
  geom_bar(alpha=0.8)+
  #geom_text(aes(label=Tumor_entity, y=8), angle=90, size=8, color=c("#333333"), fontface="plain", family = "sans", vjust=0)+
  labs(title = "", x="Patient ID", y="# peptides", fill="Transcript Type")+
  scale_y_continuous(breaks = seq(0,22,2))+
  theme(plot.title=element_text(size=24, hjust = 0.5), axis.text=element_text(size=16), axis.text.x = element_text(angle = 0),
axis.title=element_text(size=18,face="bold"), legend.text=element_text(size=16), legend.title=element_text(size=18,face="bold"))+
  theme(
    panel.grid.major = element_line(color = "grey", linetype = "dotted", size=0.8),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "transparent",colour = NA),
    plot.background = element_rect(fill = "transparent",colour = NA)
  )+
  theme(
    axis.text.x = element_text(angle = 25))
#theme_bw()
PD.2

##### 2b ## PLOT PD: Peptide distribution -- per Tumor_ID ______________________________________________________________________#####
DF.plot.2c <- fread("rawfiles/Peptides_2021_10_ErrorSamples/IN.peptides.filtered.final.all_new_perTumor.csv")
DF.plot.2c <- DF.plot.2c[1:95]
DF.plot.2c <- as.data.frame(DF.plot.2c)
DF.plot.2c$Tumor_ID = paste0(DF.plot.2c$Patient_ID,"_",DF.plot.2c$Metastasis)

ggplot(data = DF.plot.2c, aes(x=Tumor_ID))+
  geom_bar(position = "dodge", stat = "count")+
  labs(x="Tumor ID", y="Number of MS neoantigen candidates")+
  scale_y_continuous(breaks = seq(0,22,2))+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 35, size = 20),
    axis.text.y = element_text(size = 20),
    plot.title = element_blank(),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25))

##### 2c ## PLOT PD: Peptide distribution -- per Master_ID ______________________________________________________________________#####
DF.plot.2b <- DF.plot %>%
  group_by(Patient_ID, Metastasis) %>%
  mutate(N.peptides=n()) %>%
  ungroup() %>%
  select(N.peptides, everything())

ggplot(data = DF.plot.2b, aes(x=Patient_ID, y=N.peptides, fill=Metastasis))+
  geom_bar(position = "dodge", stat = "identity")+
  #geom_histogram(stat = "count", alpha=0.8)+
  labs(x="Patient ID", y="Number of MS neoantigen candidates", fill = "Tumor \nsample")+
  #scale_y_continuous(breaks = seq(0,22,2))+
  scale_y_continuous(limits = c(0, 1*max(DF.plot.2b$N.peptides)), breaks = seq(0,22,2) )+
  theme_PS()+
  #expand_limits(x=26)+
  theme(
    axis.text.x = element_text(angle = 35, size = 20),
    axis.text.y = element_text(size = 20),
    plot.title = element_blank(),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25))+
  #scale_fill_brewer(palette = "Paired")+
  #scale_fill_manual( values =c("#A6CEE3","#B2DF8A","#1F78B4", "#33A02C", "#FDBF6F", "#FF7F00"))+
  scale_fill_manual( values =c("#A6CEE3","#B2DF8A","#6BAED6", "#33A02C", "#1F78B4", "#08519C"))

## for reactive neoantigens
DF.plot.2b.reactive <- DF.plot.reactive %>%
  group_by(Patient_ID, Metastasis) %>%
  mutate(N.peptides=n()) %>%
  ungroup() %>%
  select(N.peptides, everything())

ggplot(data = DF.plot.2b.reactive, aes(x=Patient_ID, y=N.peptides, fill=Metastasis))+
  geom_bar(position = "dodge", stat = "identity")+
  #geom_histogram(stat = "count", alpha=0.8)+
  labs(x="Patient ID", y="Number of MS neoantigen candidates")+
  #scale_y_continuous(breaks = seq(0,22,2))+
  scale_y_continuous(limits = c(0, 1.3*max(DF.plot.2b$N.peptides)), breaks = seq(0,22,2) )+
  theme_PS()+
  expand_limits(x=26)+
  theme(
    axis.text.x = element_text(angle = 35))+
  #scale_fill_brewer(palette = "Paired")+
  #scale_fill_manual( values =c("#A6CEE3","#B2DF8A","#1F78B4", "#33A02C", "#FDBF6F", "#FF7F00"))+
  scale_fill_manual( values =c("#A6CEE3","#B2DF8A","#6BAED6", "#33A02C", "#1F78B4", "#08519C"))

##### 2d ## PLOT PD: Peptide distribution reactive neoantigens ____________________________________________________________######
DF.plot.tested.peptides <- data.frame (Patient_ID  = c("IN-01", "IN-04", "IN-05", "IN-11", "IN-19", "IN-22", "IN-33", "IN-37", "IN-01", "IN-04", "IN-05",
"IN-11", "IN-19", "IN-22", "IN-33", "IN-37"),
         Peptides = as.numeric(c("3", "8", "1", "2", "6", "1", "1", "1", "1", "5", "0", "3", "3", "0", "0", "7")),
         Immunogenicity = c("Reactive", "Reactive", "Reactive", "Reactive", "Reactive", "Reactive", "Reactive", "Reactive", "Non-reactive", "Non-
reactive", "Non-reactive", "Non-reactive", "Non-reactive", "Non-reactive", "Non-reactive", "Non-reactive"),
         Entity = c("Thymoma", "Renal-cell-Ca", "Leiomyosarcoma", "Pancreas-Ca \n& Endometrium-Ca", "Melanoma", "Melanoma", "Lung-Ca",
"Adeno-Ca", "", "", "", "", "", "", "", "")
)

ggplot(DF.plot.tested.peptides, aes(Patient_ID, Peptides, fill = Immunogenicity))+
  geom_bar(position = "stack", stat = "identity")+
  labs(x="Patient ID", y="Number of tested neoantigen candidates", fill = "Immunogenicity")+
  scale_y_continuous(breaks = seq(0,13,2))+
  #scale_y_continuous(breaks = c(0,1,2,3,4,5,6,7,8,9,10))+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 35, size = 20),
    axis.text.y = element_text(size = 20),
    plot.title = element_blank(),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")+
  scale_fill_manual(values=c(brewer.pal(9,"Paired")[1], brewer.pal(9,"Paired")[5]))+
  geom_text(aes(label=Entity), angle=90, size=8, color=c("#555555"), fontface="plain", family = "sans", hjust=0)

### 3 ## Gene distribution of neoantigens ______________________________________________________________________####
## 3 ## PLOT GD: Gene distribution
DF.plot.3.1 <- DF.plot %>%
  distinct(gene, Seq, Patient_ID, .keep_all = T) %>%
  group_by(gene) %>%
  summarise(count=n())
DF.plot.3.2 <- DF.plot %>%
  distinct(gene, Seq, Patient_ID, .keep_all = T)
DF.plot.3 <- merge(DF.plot.3.1, DF.plot.3.2) %>%
  filter(count>1)
GD.1 <- ggplot(data = DF.plot.3, aes(reorder(gene,desc(-count)), fill=Tumor_entity))+
  geom_bar(alpha=0.8, stat= "count")+
  #geom_text(aes(label=Tumor_entity, y=5), angle=90, size=5, fontface="italic", family = "sans", vjust=0)+
  labs(title = "Gene distribution", x="Gene", y="# peptides (different in Patient OR sequence)", color="Transcript Type")+
  scale_y_continuous(breaks = seq(0,10,2))+
  theme(plot.title=element_text(size=24, hjust = 0.5), axis.text=element_text(size=14), axis.text.x = element_text(angle = 0),
axis.title=element_text(size=18,face="bold"), legend.text=element_text(size=16), legend.title=element_text(size=18,face="bold"))+
  coord_flip()+
  theme(plot.title=element_text(size=24, hjust = 0.5), axis.text=element_text(size=16), axis.text.x = element_text(angle = 0),
axis.title=element_text(size=18,face="bold"), legend.text=element_text(size=16), legend.title=element_text(size=18,face="bold"))+
  theme(
    panel.grid.major = element_line(color = "grey", linetype = "dotted", size=0.8),
    panel.grid.minor = element_blank(),
   panel.background = element_rect(fill = "transparent",colour = NA),
   plot.background = element_rect(fill = "transparent",colour = NA),
   axis.text = element_text(size = 20),
   axis.title = element_text(size = 22),
   legend.text = element_text(size = 22),
   legend.title = element_text(size= 22,face="bold")
  )
GD.1

### 4 ## Genetic origin of neoantigens ________________________________________________________________________________________#####
## 4 a ## PLOT genetic origin: #peptides from different genebiotypes (!!!only unique Sequences!!!) ________________________#####
DF.plot.7 <- DF.plot %>%
  group_by(geneBiotype, calledBy) %>%
  summarise(N.peptides=n()) %>%
  ungroup() %>%
  mutate(geneBiotype=str_replace_all(geneBiotype, c("3prime_overlapping_ncRNA"="3'-overlapping ncRNA",
                                                    "antisense" = "lncRNA", #lncRNA (antisense)
                                                    "lincRNA" = "lncRNA",
                                                    "processed_pseudogene"="Pseudogene",
                                                    "protein_coding"="Protein Coding",
                                                    "transcribed_Processed Pseudogene"="Pseudogene",
                                                    "unProcessed Pseudogene"="Pseudogene",
                                                    "sense_intronic"="Sense Intronic",
                                                    "transcribed_Unprocessed Pseudogene;processed_transcript"="Processed Transcript",
                                                    "transcribed_Unprocessed Pseudogene"="Pseudogene",
                                                    "unitary_pseudogene"="Pseudogene",
                                                    "unPseudogene" = "Pseudogene",
                                                    "transcribed_unPseudogene" = "Pseudogene",
                                                    "transcribed_Pseudogene" = "Pseudogene",
                                                    "processed_transcript" = "Processed Transcript")))

DF.plot7.help <- DF.plot.7 %>%
  group_by(geneBiotype) %>%
  summarise(N.peptides.all=sum(N.peptides))
DF.plot.7 <- merge(DF.plot.7, DF.plot7.help)
DF.plot.7$calledBy <- sub("StrelkaRNA.Mutect2", "Mutect2_StrelkaRNA", DF.plot.7$calledBy) #Problem with "+" sign --> use "." instead which stands
for "anything"
DF.plot.7$geneBiotype <- sub("Protein Coding;Pseudogene", "Protein Coding", DF.plot.7$geneBiotype)

ggplot(data = DF.plot.7, aes(x = reorder(geneBiotype, N.peptides.all), y= N.peptides, fill = calledBy))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "", y="Number of unique MS neoantigens", x="Genetic Biotype", fill="Mutation origin")+
  theme_PS()+
  theme(legend.position = "bottom",
     axis.text.x = element_text(size = 20),
     axis.text.y = element_text(size = 22),
     plot.title = element_blank(),
     axis.title.y = element_blank(),
     axis.title.x = element_text(size=25),
     legend.text = element_text(size = 20),
     legend.title = element_text(size=25))+
  scale_x_discrete(labels=function(x){sub(";", "\n&", x)})+ # add line break at ";"
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA")) + # or change brightness etc with l=75, c=70,
  scale_y_continuous(limits = c(0, 1.05*max(DF.plot.7$N.peptides.all)), breaks = c(10,20,30,40,50, 60) )

#for reactive neoantigens
DF.plot.7.reactive <- DF.plot.reactive %>%
  group_by(geneBiotype, calledBy) %>%
  summarise(N.peptides=n()) %>%
  ungroup() %>%
  mutate(geneBiotype=str_replace_all(geneBiotype, c("3prime_overlapping_ncRNA"="3'-overlapping ncRNA",
                                                    "antisense" = "lncRNA", #lncRNA (antisense)
                                                    "lincRNA" = "lncRNA",
                                                    "protein_coding;unprocessed_pseudogene"="Protein Coding",
                                                    "processed_pseudogene"="Pseudogene",
                                                    "protein_coding"="Protein Coding",
                                                    "transcribed_Processed Pseudogene"="Pseudogene",
                                                    "unProcessed Pseudogene"="Pseudogene",
                                                    "sense_intronic"="Sense Intronic",
                                                    "transcribed_Unprocessed Pseudogene;processed_transcript"="Processed Transcript",
                                                    "transcribed_Unprocessed Pseudogene"="Pseudogene",
                                                    "unitary_pseudogene"="Pseudogene",
                                                    "unPseudogene" = "Pseudogene",
                                                    "transcribed_unPseudogene" = "Pseudogene",
                                                    "transcribed_Pseudogene" = "Pseudogene",
                         "processed_transcript" = "Processed Transcript")))

DF.plot7.help.reactive <- DF.plot.7.reactive %>%
  group_by(geneBiotype) %>%
  summarise(N.peptides.all=sum(N.peptides))
DF.plot.7.reactive <- merge(DF.plot.7.reactive, DF.plot7.help.reactive)
DF.plot.7.reactive$calledBy <- sub("StrelkaRNA.Mutect2", "Mutect2_StrelkaRNA", DF.plot.7.reactive$calledBy) #Problem with "+" sign --> use "."
instead which stands for "anything"

ggplot(data = DF.plot.7.reactive, aes(x = reorder(geneBiotype, N.peptides.all), y= N.peptides, fill = calledBy))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "", y="Number of reactive MS neoantigens", x="Genetic Biotype", fill="Mutation origin")+
  theme_PS()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 22),
        plot.title = element_blank(),
        axis.title.y = element_blank(),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size = 20),
        legend.title = element_text(size=25))+
  scale_x_discrete(labels=function(x){sub(";", "\n&", x)})+ # add line break at ";"
  scale_y_continuous(limits = c(0, 1.05*max(DF.plot.7.reactive$N.peptides.all)), breaks = c(0,2,4,6,8,10,12,14) )+
  scale_fill_manual(labels = c("DNA + RNA", "RNA"), values = c("#00BA38", "#619CFF"))

## 4 b ## PLOT mutation type: #peptides from different mutation types (!!!only unique Sequences!!!) __________ #######
# Mutation type for all neoantigens
DF.plot.7.2 <- DF.plot %>%
  group_by(EFFECT, calledBy, transcriptTypes) %>%
  summarise(N.peptides=n()) %>%
  ungroup() %>%
  mutate(EFFECT=str_replace_all(EFFECT, c("missense_variant;non_coding_transcript_exon_variant"="Coding missense variant",
                    "non_coding_transcript_exon_variant"="Non-coding missense variant",
                         "splice_acceptor_variant-intron_variant" = "Splice-site & intron variant",
                         "splice_donor_variant-intron_variant" = "Splice-site & intron variant",
                         "missense_variant"="Coding missense variant",
                         "frameshift_variant"="Frameshift variant",
                         "missense_variant;non_coding_transcript_exon_variant"="Coding missense variant",
                         "splice_acceptor_variant&intron_variant"="Splice-site & intron variant",
                         "splice_donor_variant&intron_variant"="Splice-site & intron variant")))
DF.plot7.2.help <- DF.plot.7.2 %>%
  group_by(EFFECT) %>%
  summarise(N.peptides.all=sum(N.peptides))
DF.plot.7.2 <- merge(DF.plot.7.2, DF.plot7.2.help)
DF.plot.7.2$calledBy <- sub("StrelkaRNA.Mutect2", "Mutect2_StrelkaRNA", DF.plot.7.2$calledBy) #Problem with "+" sign --> use "." instead which
stands for "anything"

ggplot(data = DF.plot.7.2, aes(x = reorder(EFFECT, N.peptides.all), y= N.peptides, fill = calledBy))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "", y="Number of unique MS neoantigens", x="Mutational effect", fill="Mutation origin")+
  theme_PS()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 22),
        plot.title = element_blank(),
        axis.title.y = element_blank(),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size = 20),
        legend.title = element_text(size=25))+
  scale_x_discrete(labels=function(x){sub(" ", "\n", x)})+ # add line break at "blank"
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA")) + # or change brightness etc with l=75, c=70,
  scale_y_continuous(limits = c(0, 1.05*max(DF.plot.7.2$N.peptides.all)), breaks = c(0,10,20,30) )

# Mutation Type for reactive neoantigens
DF.plot.7.2.reactive <- DF.plot.reactive %>%
  group_by(EFFECT, calledBy, transcriptTypes) %>%
  summarise(N.peptides=n()) %>%
  ungroup() %>%
  mutate(EFFECT=str_replace_all(EFFECT, c("missense_variant;non_coding_transcript_exon_variant"="Coding missense variant",
                    "non_coding_transcript_exon_variant"="Non-coding missense variant",
                    "splice_acceptor_variant-intron_variant" = "Splice-site & intron variant",
                    "splice_donor_variant-intron_variant" = "Splice-site & intron variant",
                    "missense_variant"="Coding missense variant",
                    "frameshift_variant"="Frameshift variant",
                                    "missense_variant;non_coding_transcript_exon_variant"="Coding missense variant",
                                    "splice_acceptor_variant&intron_variant"="Splice-site & intron variant",
                                    "splice_donor_variant&intron_variant"="Splice-site & intron variant")))
DF.plot7.2.reactive.help <- DF.plot.7.2.reactive %>%
  group_by(EFFECT) %>%
  summarise(N.peptides.all=sum(N.peptides))
DF.plot.7.2.reactive <- merge(DF.plot.7.2.reactive, DF.plot7.2.reactive.help)
DF.plot.7.2.reactive$calledBy <- sub("StrelkaRNA.Mutect2", "Mutect2_StrelkaRNA", DF.plot.7.2.reactive$calledBy) #Problem with "+" sign --> use "."
instead which stands for "anything"

ggplot(data = DF.plot.7.2.reactive, aes(x = reorder(EFFECT, N.peptides.all), y= N.peptides, fill = calledBy))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "", y="Number of reactive MS neoantigens", x="Mutational effect", fill="Mutation origin")+
  theme_PS()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 22),
        plot.title = element_blank(),
        axis.title.y = element_blank(),
        axis.title.x = element_text(size=25),
        legend.text = element_text(size = 20),
        legend.title = element_text(size=25))+
  scale_x_discrete(labels=function(x){sub(" ", "\n", x)})+ # add line break at "blank"
  scale_fill_manual(labels = c("DNA + RNA", "RNA"), values = c("#00BA38", "#619CFF"))+
  scale_y_continuous(limits = c(0, 1.05*max(DF.plot.7.2.reactive$N.peptides.all)), breaks = c(0,2,4, 6, 8,10) )


### 5 ## Quality Control ____________________________________________________________________________________####
## 5 - 1 ## QC RNA variants: coverage, allele frequency and A-G mut
DF.plot.QC <- fread("rawfiles/Peptides_2021_10_ErrorSamples/IN.peptides.filtered.final.all_new_QAassess.csv")
DF.plot.QC <- DF.plot.QC[1:91]
DF.plot.QC <- as.data.frame(DF.plot.QC)
DF.plot.QC <- DF.plot.QC[DF.plot.QC$calledBy == "StrelkaRNA",]

ggplot(DF.plot.QC, aes(x= TumorCoverage_RNA, y= TumorCoverage_DNA))+
  geom_point()+
  #geom_jitter()+
  scale_y_log10()+
  scale_x_log10()+
  geom_hline(yintercept=10, linetype="dashed", color = "red")

DF.plot.QC$Mutation_ID <- paste(DF.plot.QC$Chrom, DF.plot.QC$Pos,DF.plot.QC$Ref_Alt, sep = "_")
DF.plot.QC$transcriptTypes[DF.plot.QC$transcriptTypes == "coding+noncoding"] <- "coding"

#add DNA coverage info per Mutation_ID and filter
setDT(DF.plot.QC)[ , DNACoverage := all.patients_DF_uniqueMut_DNAcoverage$TumorCoverage.Mutect2.filtered[match(DF.plot.QC$Mutation_ID ,
all.patients_DF_uniqueMut_DNAcoverage$Mutation_ID)] , ]
DF.plot.QC$DNACoverage_short <- DF.plot.QC$DNACoverage
DF.plot.QC$DNACoverage_short[DF.plot.QC$DNACoverage_short < 3] <- "no"
DF.plot.QC$DNACoverage_short[DF.plot.QC$DNACoverage >= 3] <- "yes"

DF.plot.QC[, .N, by=.(DNACoverage_short)]

#only reactive neoantigens
DF.plot.QC.reactive <- DF.plot.QC[DF.plot.QC$Seq_ID_short %in% reactive_neoantigens,]

#plot
ggplot(DF.plot.QC[DF.plot.QC$mutationType == "substitution",], aes(x=Ref_Alt2, fill = DNACoverage_short))+
  geom_bar(stat = "count")+
  #facet_wrap(~transcriptTypes)+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  labs(x="Nucleic acid changes Ref_Alt", y="Number of RNA peptides", fill ="Coverage on DNA above 5 reads")+
  #scale_fill_manual(labels = c("RNA"), values = c("#619CFF"))+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    strip.text.x = element_text(size = 25, face = "bold"),
    axis.text.y = element_text(size = 25, vjust = 0.8),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")

DF.plot.QC[, .N, by=.(Ref_Alt2)]
```

```
ggplot(DF.plot.QC, aes(x=TumorCoverage_DNA, y= DNACoverage))+
  geom_point()

## 5 - 2 ## QC RNA variants: Database check
#setDT(DF.plot.QC)[ , RNAedit.RADAR := all.patients_DF_uniqueMut_DNAcoverage$RNAedit.RADAR[match(DF.plot.QC$Mutation_ID ,
all.patients_DF_uniqueMut_DNAcoverage$Mutation_ID)] , ]
setDT(DF.plot.QC)[ , RNAedit.REDI := all.patients_DF_uniqueMut_DNAcoverage$RNAedit.REDI[match(DF.plot.QC$Mutation_ID ,
all.patients_DF_uniqueMut_DNAcoverage$Mutation_ID)] , ]

#DF.plot.QC$Database <- paste(DF.plot.QC$RNAedit.RADAR, DF.plot.QC$RNAedit.REDI, sep = "_")
#DF.plot.QC[Database %in% c("yes_yes"), Database := "both"]
#DF.plot.QC[Database %in% c("no_no"), Database := "none"]
#DF.plot.QC[Database %in% c("no_yes"), Database := "RNAedit.REDI"]
#DF.plot.QC[Database %in% c("yes_no"), Database := "RNAedit.RADAR"]
#DF.plot.QC[Database %in% c("NA_NA"), Database := "NA"]

ggplot(DF.plot.QC[DF.plot.QC$mutationType == "substitution",], aes(x=Ref_Alt2, fill = RNAedit.REDI))+
  geom_bar(stat = "count")+
  #facet_wrap(~transcriptTypes)+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
  labs(x="Nucleic acid changes Ref_Alt", y="Number of RNA peptides")+
  scale_fill_brewer(palette = "Set1")+
  #scale_fill_manual(labels = c("RNA"), values = c("#619CFF"))+
  theme(
    axis.text.x = element_text(angle = 45, size = 25, vjust = 1, hjust=1),
    plot.title = element_blank(),
    strip.text.x = element_text(size = 25, face = "bold"),
    axis.text.y = element_text(size = 25, vjust = 0.8),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")

ggplot(DF.plot.QC, aes(x=Database, y= DNACoverage))+
  geom_point()+
  geom_hline(yintercept=50, linetype="dashed", color = "red")

## 5 - 3 ## PLOT QC prediction: KD_best vs. Score-MS
DF.plot.8 <- DF.plot %>%
  mutate(BA.best= (BA.best.MHCflurry+BA.best.netMHC)/2)
ggplot(data=DF.plot.8, mapping=aes(y=BA.best.MHCflurry, x=scoreMS_pFind))+
  #geom_point(aes(color=allele.best.BA.MHCflurry), size=7)+
  geom_jitter(aes(size=allele.frequency.MHCflurry, color=allele.best.BA.MHCflurry))+
  scale_size(range=c(1,9))+
  labs(title = "", x="MS score", y="Kd (best allele with MHCflurry) [nM]")+
  theme_PS()+
  coord_cartesian(xlim = c(0,0.6))+
  scale_x_continuous(breaks = seq(0, 0.6, 0.1))+
  scale_y_log10()

### 6  ## Comparison In and Mel15_T1 neoantigens ___________________________________________________________________________#####
DF.plot.Mel15 <- fread("rawfiles/Peptides_2021/neoantigens_IN_Mel15_comb.csv")

ggplot(DF.plot.Mel15, aes(Tumor_ID, neoantigens, fill = calledBY))+
  geom_bar(stat = "identity")+
  theme_PS()+
  theme(
    axis.text.x = element_text(angle = 45, size = 20, vjust = 1, hjust=1),
    plot.title = element_blank(),
    axis.text.y = element_text(size = 20),
    axis.title.y = element_text(size=25),
    axis.title.x = element_text(size=25),
    legend.text = element_text(size = 20),
    legend.title = element_text(size=25),
    legend.position = "bottom")+
  #guides (fill = guide_legend(ncol = 1))+
  scale_fill_hue(labels = c("DNA", "DNA + RNA", "RNA"))+
  labs(x="Patient ID", y="Number of neoantigen candidates", fill="Mutation Origin")
```

6.9.5 Analysis of immunogenicity assessment data

```
library(data.table)
library(ggplot2)
library(scales)
library(ggpubr)
library(readr)
library(xlsx)
library(readxl)
library(writexl)
library(tidyr)
library(splitstackshape)
library(dplyr)
library(ggrepel)

# __________ (1) ImmuNEO new/final peptide cohort _______________________________________________________####
##  _______________ data analysis ______________________________________________________________####
acDC_data <- fread("Summary_acDCs_all_V1.csv" )

# calculate the deltas for the mutated peptide to all controls
acDC_data[, Delta_irrel := mut_pep - irrel_pep]
acDC_data[, Delta_unplused := mut_pep - unpulsed]
acDC_data[, Delta_all := mut_pep - ((irrel_pep + unpulsed)/2)]

# calculate the ratios for the mutated peptide to all controls
acDC_data[, Ratio_irrel := mut_pep / irrel_pep]
acDC_data[, Ratio_unplused := mut_pep / unpulsed]
acDC_data[, Ratio_all := mut_pep / ((irrel_pep + unpulsed)/2)]

acDC_data[ ,Peptide_ID_2 := gsub(" ", "_", Peptide_ID_2)]
acDC_data[ ,Sample_type := gsub("FT", "non-enriched", Sample_type)]
acDC_data[ ,Sample_type := gsub("normal", "non-enriched", Sample_type)]
#colnames(acDC_data) <- gsub("Peptide ID", "Peptide_ID", colnames(acDC_data))

# for IN-4 add values from Ratio and Delta mut/irrel. to Ratio and Delta all (because only mutated and irrelevant pulsed conditions in triplicates)
#acDC_data$Ratio_all <- ifelse(is.na(acDC_data$Ratio_all), acDC_data$Ratio_irrel, acDC_data$Ratio_all)
#acDC_data$Delta_all <- ifelse(is.na(acDC_data$Delta_all), acDC_data$Delta_irrel, acDC_data$Delta_all)

acDC_data[acDC_data < 0] <- 0   # adjust negative values --> set to 0
acDC_data[acDC_data == "NaN"] <- 0
acDC_data[acDC_data == "Inf"] <- 0

### ________________ plots ____________________________________________________________________####
#### ____________________ very general plot _________________________________________________________####
ggplot(acDC_data, aes(x=Ratio_all, y=Delta_all, colour = Patient_ID))+
  geom_point()+
  labs(title="Immunogenicity assays neoantigen candidates first cohort",
       x ="Ratio of mean spots from mutated peptides vs. controls", y = "Difference of mean spots from mutated peptides vs. controls") +
  guides(colour=guide_legend(ncol=2))

# plot with interesting data points labeled
ggplot(acDC_data, aes(x=Ratio_all, y=Delta_all, colour = Patient_ID,label = Peptide_ID))+
  geom_point(size = 2,alpha = 0.6)+
  labs(title="Immunogenicity assays neoantigen candidates first cohort",
       x ="Ratio of mean spots from mutated peptides vs. controls", y = "Difference of mean spots from mutated peptides vs. controls") +
  guides(colour=guide_legend(ncol=2))+
  geom_text(aes(label=ifelse(Delta_all>50 & Ratio_all> 2,as.character(Peptide_ID),'')),hjust=0,vjust=0, nudge_x = 0.1, nudge_y = 0.1)

### ____________________________ compare mutated vs. mean of all controls ______________________________________________####
####_________________________________ all data points _________________________________________________________________#####
acDC_data_highlights <- acDC_data[ Delta_all>50 & Ratio_all >2] # select for criteria
acDC_data_highlights

ggplot(acDC_data, aes(x=Ratio_all, y=Delta_all))+
  geom_point(size = 2,alpha = 0.6)+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
       x ="Ratio of mean spots from mutated peptides vs. controls", y = "Difference of mean spots from mutated peptides vs. controls") +
  guides(colour=guide_legend(ncol=1))+
  theme(axis.text.x = element_text(size= 20),
        axis.text.y = element_text(size= 20),
        plot.title = element_text(size= 15),
        axis.title.y = element_text(size=15),
        axis.title.x = element_text(size=15),
        legend.text = element_text(size= 15),
        legend.title = element_text(size=15, face = "bold"),
        panel.background = element_rect(fill="white"),
        panel.grid.minor.y = element_line(size=3),
        panel.grid.major = element_line(colour = "grey"),
        plot.background = element_rect(fill="white")) +
  geom_point(data = acDC_data_highlights,aes(x=Ratio_all, y=Delta_all, colour = Peptide_ID)) # add another layer of coloured points ontop of general plot

ggplot(acDC_data, aes(x=Ratio_all, y=Delta_all))+
  geom_point(size = 2)+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
       x ="Ratio of mean spots from mutated peptides vs. controls", y = "Difference of mean spots from mutated peptides vs. controls") +
  guides(colour=guide_legend(ncol=1))+
  theme_light(base_size = 20) +
  geom_point(data = acDC_data_highlights,
             aes(x=Ratio_all, y=Delta_all, colour = Peptide_ID_2, shape = Biotype), size=3) #add another layer of coloured points ontop of general plot and adapt size of these points

# only plot interesting peptides
ggplot(acDC_data_highlights, aes(x=Ratio_all, y=Delta_all, colour = Peptide_ID ))+
  geom_point(size = 3,alpha = 0.6, aes(shape = Sample_type))+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
       x ="Ratio of mean spots from mutated peptides vs. controls", y = "Difference of mean spots from mutated peptides vs. controls") +
  guides(colour=guide_legend(ncol=1))+
  theme_light(base_size = 20)+
  geom_text_repel(data=filter(acDC_data_highlights, Delta_all>50 & Ratio_all> 2), aes(label=Peptide_ID))

#### ______________________________ compare mutated vs. mean irrelevant pulsed ________________________________________________####
####___________________________________ all data points _______________________________________________________________________#####
acDC_data_highlights <- acDC_data[ Delta_irrel >= 50 & Ratio_irrel >=2] # select for criteria
reactive_neoantigens <- unique(acDC_data_highlights$Peptide_ID)

ggplot(acDC_data, aes(x=Ratio_irrel, y=Delta_irrel))+
  geom_point(size = 2)+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
       x ="Ratio of mean spots from mutated vs. irrelevant peptides", y = "Difference of mean spots from mutated vs. irrelevant peptides") +
  guides(colour=guide_legend(ncol=1, "Peptide ID"))+
  theme_light(base_size = 25) +
  theme(plot.title = element_blank())+
  facet_wrap(~ Biotype, scales = "free")+
  coord_flip()+
  geom_hline(yintercept=50, linetype="dashed", color = "red")+
  geom_vline(xintercept=2, linetype="dashed", color = "red")+
  #geom_text_repel(data = acDC_data_highlights, aes(label=Peptide_ID),hjust=1,vjust=0, nudge_x = 0.1, nudge_y = 0.1)+
  geom_point(data = acDC_data_highlights,
             aes(x=Ratio_irrel, y=Delta_irrel, colour = Peptide_ID, shape = Sample_type), size=5)  #add another layer of coloured points ontop of general plot and adapt size of these points

## Option 2 seperate by PBMC and TILs
#eclude olf IN-04 Experiment
acDC_data_revision <- acDC_data[Experiment != "acDC04#1"]
acDC_data_highlights_revision <- acDC_data_highlights[Experiment != "acDC04#1"]

ggplot(acDC_data_revision[acDC_data_revision$Biotype == "PBMC"], aes(x=Ratio_irrel, y=Delta_irrel))+
  geom_point(size = 2)+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
       x ="Ratio of mean SFU from mutated vs. irrelevant peptides", y = "Difference of mean SFU from mutated vs. irrelevant peptides") +
  guides(colour=guide_legend(ncol=1, "Peptide ID"))+
  theme_light(base_size = 25) +
  theme(plot.title = element_blank())+
  #theme(legend.position = "none")+
  #facet_wrap(~ Sample_type, scales = "free")+
  coord_flip()+
  geom_hline(yintercept=50, linetype="dashed", color = "red")+
  geom_vline(xintercept=2, linetype="dashed", color = "red")+
  #geom_text_repel(data = acDC_data_highlights_revision, aes(label=Experiment_time),hjust=1,vjust=0, nudge_x = 0.1, nudge_y = 0.1)+
  geom_point(data = acDC_data_highlights_revision[acDC_data_highlights_revision$Biotype == "PBMC"],
             aes(x=Ratio_irrel, y=Delta_irrel, colour = Peptide_ID, shape = Sample_type), size=5) + #add another layer of coloured points ontop of general plot and adapt size of these points
  scale_shape_manual(values=c(15, 17))

ggplot(acDC_data[acDC_data$Biotype == "TIL"], aes(x=Ratio_irrel, y=Delta_irrel))+
  geom_point(size = 2)+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
       x ="Ratio of mean SFU from mutated vs. irrelevant peptides", y = "Difference of mean SFU from mutated vs. irrelevant peptides") +
  guides(colour=guide_legend(ncol=1, "Peptide ID"))+
  theme_light(base_size = 25) +
  theme(plot.title = element_blank())+
  facet_wrap(~ Sample_type)+
  coord_flip()+
  geom_hline(yintercept=50, linetype="dashed", color = "red")+
  geom_vline(xintercept=2, linetype="dashed", color = "red")+
  #geom_text_repel(data = acDC_data_highlights, aes(label=Peptide_ID),hjust=1,vjust=0, nudge_x = 0.1, nudge_y = 0.1)+
  geom_point(data = acDC_data_highlights[acDC_data_highlights$Biotype == "TIL"],
           aes(x=Ratio_irrel, y=Delta_irrel, colour = Peptide_ID), size=5)  #add another layer of coloured points ontop of general plot and adapt size of
these points

## change axis to better view positive points
ggplot(acDC_data, aes(x=Ratio_irrel, y=Delta_irrel))+
  geom_point(size = 2)+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
      x ="Ratio of mean spots from mutated vs. irrelevant peptides", y = "Difference of mean spots from mutated vs. irrelevant peptides") +
  guides(colour=guide_legend(ncol=1, "Peptide ID"))+
  theme_light(base_size = 20) +
  facet_wrap(~ Biotype, scales = "free")+
  scale_x_log10()+
  scale_y_log10()+
  coord_flip()+
  geom_text_repel(data = acDC_data_highlights, aes(label=Peptide_ID),hjust=1,vjust=0, nudge_x = 0.1, nudge_y = 0.1)+
  geom_point(data = acDC_data_highlights,
           aes(x=Ratio_irrel, y=Delta_irrel, colour = Peptide_ID, shape = Sample_type), size=3)  #add another layer of coloured points ontop of general
plot and adapt size of these points

## without ImmuNEO-4 because maybe wrong positives?!
ggplot(acDC_data[Experiment != "acDC04#1"], aes(x=Ratio_irrel, y=Delta_irrel))+
  geom_point(size = 2)+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
      x ="Ratio of mean spots from mutated vs. irrelevant peptides", y = "Difference of mean spots from mutated vs. irrelevant peptides") +
  guides(colour=guide_legend(ncol=1, "Peptide ID"))+
  theme_light(base_size = 20) +
  facet_wrap(~ Biotype, scales = "free")+
  geom_point(data = acDC_data_highlights[Experiment != "acDC04#1"],
           aes(x=Ratio_irrel, y=Delta_irrel, colour = Peptide_ID, shape = Sample_type), size=3)  #add another layer of coloured points ontop of general
plot and adapt size of these points

# ________ (2) Healthy donor ____________________________________________________________________________#####
acDC_data_HD <- fread("acDC Johannes Healthy Donor.csv" )

# calculate the deltas for the mutated peptide to all controls
acDC_data_HD[, Delta_irrel := mut_pep - irrel_pep]
#acDC_data_HD[, Delta_unplused := mut_pep - unpulsed]
#acDC_data_HD[, Delta_all := mut_pep - ((irrel_pep + unpulsed)/2)]

# calculate the ratios for the mutated peptide to all controls
acDC_data_HD[, Ratio_irrel := mut_pep / irrel_pep]
acDC_data_HD[, Ratio_unplused := mut_pep / unpulsed]
acDC_data_HD[, Ratio_all := mut_pep / ((irrel_pep + unpulsed)/2)]

acDC_data_HD[ ,Peptide_ID_2 := gsub(" ", "_", Peptide_ID_2)]
acDC_data_HD[ ,Sample_type := gsub("FT", "non-enriched", Sample_type)]
acDC_data_HD[ ,Sample_type := gsub("normal", "non-enriched", Sample_type)]
colnames(acDC_data_HD) <- gsub("Peptide ID", "Peptide_ID", colnames(acDC_data_HD))

# for IN-4 add values from Ratio and Delta mut/irrel. to Ratio and Delta all (because only mutated and irrelevant pulsed conditions in triplicates)
#acDC_data_HD$Ratio_all <- ifelse(is.na(acDC_data_HD$Ratio_all), acDC_data_HD$Ratio_irrel, acDC_data_HD$Ratio_all)
#acDC_data_HD$Delta_all <- ifelse(is.na(acDC_data_HD$Delta_all), acDC_data_HD$Delta_irrel, acDC_data_HD$Delta_all)

acDC_data_HD[acDC_data_HD < 0] <- 0   # adjust negative values --> set to 0
acDC_data_HD[acDC_data_HD == "NaN"] <- 0
acDC_data_HD[acDC_data_HD == "Inf"] <- 0
acDC_data_HD <- acDC_data_HD[Sample_type != "plus naive"]

#### ______________________________ compare mutated vs. mean irrelevant pulsed ______________________####
####___________________________________ all data points ____________________________________________________________________#####
acDC_data_HD_highlights <- acDC_data_HD[ Delta_irrel >= 50 & Ratio_irrel >=2] # select for criteria
reactive_neoantigens <- unique(acDC_data_highlights$Peptide_ID)

ggplot(acDC_data_HD, aes(x=Ratio_irrel, y=Delta_irrel))+
  geom_point(size = 2)+
  labs(title="Immunogenicity assays neoantigen candidates final cohort",
      x ="Ratio of mean spots from mutated vs. irrelevant peptides", y = "Difference of mean spots from mutated vs. irrelevant peptides") +
  guides(colour=guide_legend(ncol=1, "Peptide ID"))+
  theme_light(base_size = 20) +
  theme(plot.title = element_blank())+
  facet_wrap(~ Biotype, scales = "free")+
  scale_y_continuous(limits = c(0,400))+
  scale_x_continuous(limits = c(0,7))+
  coord_flip()+
  geom_hline(yintercept=50, linetype="dashed", color = "red")+
  geom_vline(xintercept=2, linetype="dashed", color = "red")+
  geom_point(data = acDC_data_HD_highlights,
             aes(x=Ratio_irrel, y=Delta_irrel, colour = Peptide_ID, shape = Donor_ID), size=5)+ #add another layer of coloured points ontop of general plot and adapt size of these points
  scale_shape_manual(values=c(15, 16, 17, 18))
```

6.9.6 Analysis of flow-cytometry phenotyping data

```
library(data.table)
library(csv)
library(readxl)
library(writexl)
library(tidyr)
library(splitstackshape)
library(readr)
library(ggplot2)
library(dplyr)
library(scales)
library(xlsx)
library(pheatmap)

##______ 1. load data _______________________________________________________________________________########################
phenotyping_table<- fread("ImmuNEO-15.2_P1-2_export.csv") %>% as.data.frame()
row.names(phenotyping_table) <- phenotyping_table$V1
phenotyping_table$V1 <- NULL
phenotyping_table <- data.table(phenotyping_table, keep.rownames = TRUE)

##______ 2. clean the data __________________________________________________________________________########################
# columns
colnames(phenotyping_table) <- sub("Size/Alive/Single Cells1/", "", colnames(phenotyping_table)) #remove too long column names
colnames(phenotyping_table) <- gsub("Single Cells2", "Singles", colnames(phenotyping_table))
colnames(phenotyping_table) <- gsub(" ", "", colnames(phenotyping_table)) # delete all blank spaces
colnames(phenotyping_table) <- gsub("\\/", "_", colnames(phenotyping_table)) # replace "." with "_"
colnames(phenotyping_table) <- gsub("\\|Count", "", colnames(phenotyping_table))
#colnames(dt) <- gsub("\\.\\.+", ".", colnames(dt)) # replace multiple .... with one .

# rows
phenotyping_table[ ,rn := gsub(" ", "_", rn)] # replace blank space with "_" in row names
phenotyping_table[ ,rn := gsub("\\.fcs$", "", rn)]
#phenotyping_table[ ,rn := gsub("sample_2", "sample_15-2", rn)]
phenotyping_table <- phenotyping_table[!rn %in% c("Mean", "SD")]

#values
phenotyping_table[phenotyping_table == ""] <- NA
phenotyping_table[phenotyping_table == "NA"] <- NA
phenotyping_table[phenotyping_table == " "] <- NA
phenotyping_table <- phenotyping_table[, lapply(.SD, function(x){gsub(",0|,00","",x)})] # iterate over all columns and remove every finding of ,0 OR ,00
phenotyping_table[, Singles := gsub(",", ".", Singles)] # convert possible non-numeric numbers into a clean number step 1
phenotyping_table <- phenotyping_table[, lapply(.SD, as.numeric), by=rn] # convert possible non-numeric numbers into a clean number step 2

if(any(duplicated(colnames(phenotyping_table)))){
  stop("DUPLICATED CELL")
}

#split the row names
phenotyping_analysis <- phenotyping_table[, .(rn, Singles)]
phenotyping_analysis <- cSplit(phenotyping_analysis, "rn", "_")
phenotyping_analysis <- cSplit(phenotyping_analysis, "rn_4", "-")
phenotyping_analysis <- phenotyping_analysis[, .(patient=rn_2, panelID=rn_4_1, panelSub=rn_4_2, singles=Singles)]

#splitting when different panle structure
#phenotyping_analysis <- phenotyping_table[, .(rn, Singles)]
#phenotyping_analysis <- cSplit(phenotyping_analysis, "rn", "_")
#phenotyping_analysis <- phenotyping_analysis[1]
#phenotyping_analysis <- phenotyping_analysis[, .(patient=rn_2, panelID=rn_4, singles=Singles)]
#phenotyping_analysis[, panelID := gsub("general", "1", panelID)]

##______ 3. Analysis ________________________________________________________________________________########################
#function for calculation the sums of counts for specific columns defined in the coming script
getRowSums <- function(ColText, ColName){
  foundcols <- grep(ColText, colnames(phenotyping_table)) # grep that word in all column names
  tmp <- phenotyping_table[, foundcols, with=F] # select subset of columns
  NArow <- which(!rowSums(!is.na(tmp)))
  rsums <- rowSums(tmp, na.rm = T) # get rowsums
  rsums[NArow] <- NA
  phenotyping_analysis[, eval(ColName) := rsums] # assign rowsums to analysis data
  #print(phenotyping_analysis)
}

##______________________ General cell types ______________________________________________________________#######################
ColText <- "Lymphocytes$"
ColName <- "lymphocytes_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD45\\+$"
ColName <- "CD45_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3\\+$"
ColName <- "CD3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

##______________________ CD4 T cells ____________________________________________________________________#######################
# for each row calculate the sum of the values from the columns that contain ... and add it to a new data table
#   - "CD4+"
#   - "CD4+_CD62L+CD45RA+" and name it CD4_Tn
#   - "CD4+_CD62L+CD45RA-" and name it CD4_Tcm
#   - "CD4+_CD62L-CD45RA+" and name it CD4_Teff
#   - "CD4+_CD62L-CD45RA-" and name it CD4_Tem
ColText <- "CD4\\+"
ColName <- "CD4_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA\\+"
ColName <- "CD4_Tn_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA-"
ColName <- "CD4_Tcm_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA\\+"
ColName <- "CD4_Teff_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA-"
ColName <- "CD4_Tem_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD103\\+"
ColName <- "CD4_Trm_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD25\\+CD127low"
ColName <- "CD4_Tregs_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#____ CD4 cells and inhibitory markers
#      - "CD4+" & "PD-1+_LAG-3+TIM-3+" and name it CD4_exhaustion3
#      - "CD4+" & "PD-1+_LAG-3+TIM-3-" or "PD-1+_LAG-3-TIM-3+" or "PD-1-_LAG-3+TIM-3+" and name it CD4_inhibition2
#      - "CD4+" & "PD-1+_LAG-3-TIM-3-" & "PD-1-_LAG-3+TIM-3-" or "PD-1-_LAG-3-TIM-3+" and name it CD4_inhibition1
#      - "CD4+" & "PD-1-_LAG-3-TIM-3-" and name it CD4_inhibition0
ColText <- "CD4\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD4_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD4_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)
```

```
#____ CD4 cells and activation markers
ColText <- "CD4\\+.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD4_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+|HLA-DR\\+$)"
ColName <- "CD4_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+.*(HLA-DR-CD103-|CD103-HLA-DR-|HLA-DR-$)"
ColName <- "CD4_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)



#__ Naive CD4 T cells and inhibitory marker and activation marker
#     - "CD4+_CD62L+CD45RA+" & "PD-1+_LAG-3+TIM-3+" and name it CD4_naive_exhaustion3
#     - "CD4+_CD62L+CD45RA+" & "PD-1+_LAG-3+TIM-3-" or "PD-1+_LAG-3-TIM-3+" or "PD-1-_LAG-3+TIM-3+" and name it CD4_naive_exhaustion2
#     - "CD4+_CD62L+CD45RA+" & "PD-1+_LAG-3-TIM-3-" & "PD-1-_LAG-3+TIM-3-" or "PD-1-_LAG-3-TIM-3+" and name it CD4_naive_exhaustion1
#     - "CD4+_CD62L+CD45RA+" & "PD-1-_LAG-3-_IM-3-" and name it CD4_naive_exhaustion0
ColText <- "CD4\\+_CD62L\\+CD45RA\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Tn_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Tn_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD4_Tn_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD4_Tn_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA\\+.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD4_Tn_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA\\+.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+)"
ColName <- "CD4_Tn_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA\\+.*(HLA-DR-CD103-|CD103-HLA-DR-)"
ColName <- "CD4_Tn_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#__ CD4 Tcm cells and marker expression
ColText <- "CD4\\+_CD62L\\+CD45RA-.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Tcm_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA-.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Tcm_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA-.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD4_Tcm_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA-.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD4_Tcm_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA-.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD4_Tcm_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA-.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+)"
ColName <- "CD4_Tcm_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L\\+CD45RA-.*(HLA-DR-CD103-|CD103-HLA-DR-)"
ColName <- "CD4_Tcm_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)
```

```
#__ CD4 T effector cells and marker expression
ColText <- "CD4\\+_CD62L-CD45RA\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Teff_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Teff_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD4_Teff_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD4_Teff_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA\\+.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD4_Teff_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA\\+.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+)"
ColName <- "CD4_Teff_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA\\+.*(HLA-DR-CD103-|CD103-HLA-DR-)"
ColName <- "CD4_Teff_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#__ CD4 Tem cells and marker expression
ColText <- "CD4\\+_CD62L-CD45RA-.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Tem_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA-.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Tem_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA-.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD4_Tem_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA-.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD4_Tem_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA-.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD4_Tem_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA-.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+)"
ColName <- "CD4_Tem_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD45RA-.*(HLA-DR-CD103-|CD103-HLA-DR-)"
ColName <- "CD4_Tem_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#__ CD4 Tregs cells and inhibitory markers
ColText <- "CD25\\+CD127low.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Tregs_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD25\\+CD127low.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD4_Tregs_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD25\\+CD127low.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD4_Tregs_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD25\\+CD127low.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD4_Tregs_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)
```

```
#__ CD4 Trm cells and activation markers
ColText <- "CD4\\+_CD62L-CD103\\+.*(HLA-DR\\+$)"
ColName <- "CD4_Trm_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD4\\+_CD62L-CD103\\+.*(HLA-DR-$)"
ColName <- "CD4_Trm_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

##_______________________ CD8 T cells _______________________________________________________________________########################
#_____CD8 T cells and subsets Panel 1-1 and Panel 1-2
#    - "CD8+"
#    - "CD8+_CD62L+CD45RA+" and name it CD8_Tn
#    - "CD8+_CD62L+CD45RA-" and name it CD8_Tcm
#    - "CD8+_CD62L-CD45RA+" and name it CD8_Teff
#    - "CD8+_CD62L-CD45RA-" and name it CD8_Tem
#    - "CD8+_CD62L-CD103+" and name it CD8_Trm

ColText <- "CD8\\+"
ColName <- "CD8_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA\\+"
ColName <- "CD8_Tn_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA-"
ColName <- "CD8_Tcm_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA\\+"
ColName <- "CD8_Teff_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA-"
ColName <- "CD8_Tem_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD103\\+"
ColName <- "CD8_Trm_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#____ CD8 cells and inhibitory markers
ColText <- "CD8\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD8_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD8_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#____ CD8 cells and activation markers
ColText <- "CD8\\+.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD8_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+|HLA-DR\\+$)"
ColName <- "CD8_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+.*(HLA-DR-CD103-|CD103-HLA-DR-|HLA-DR-$)"
ColName <- "CD8_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#__ Naive CD8 T cells and marker expression
ColText <- "CD8\\+_CD62L\\+CD45RA\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Tn_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)
```

```
ColText <- "CD8\\+_CD62L\\+CD45RA\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Tn_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD8_Tn_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD8_Tn_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA\\+.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD8_Tn_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA\\+.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+)"
ColName <- "CD8_Tn_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA\\+.*(HLA-DR-CD103-|CD103-HLA-DR-)"
ColName <- "CD8_Tn_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#__ CD8 Tcm cells and marker expression
ColText <- "CD8\\+_CD62L\\+CD45RA-.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Tcm_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA-.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Tcm_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA-.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD8_Tcm_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA-.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD8_Tcm_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA-.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD8_Tcm_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA-.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+)"
ColName <- "CD8_Tcm_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L\\+CD45RA-.*(HLA-DR-CD103-|CD103-HLA-DR-)"
ColName <- "CD8_Tcm_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#__ CD8 T effector cells and markers expression
ColText <- "CD8\\+_CD62L-CD45RA\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Teff_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Teff_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD8_Teff_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD8_Teff_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA\\+.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD8_Teff_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA\\+.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+)"
```

```
ColName <- "CD8_Teff_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA\\+.*(HLA-DR-CD103-|CD103-HLA-DR-)"
ColName <- "CD8_Teff_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#__ CD8 Tem cells and markers expression
ColText <- "CD8\\+_CD62L-CD45RA-.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Tem_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA-.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "CD8_Tem_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA-.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "CD8_Tem_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA-.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "CD8_Tem_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA-.*(HLA-DR\\+CD103\\+|CD103\\+HLA-DR\\+)"
ColName <- "CD8_Tem_Activ2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA-.*(HLA-DR\\+CD103-|HLA-DR-CD103\\+|CD103\\+HLA-DR-|CD103-HLA-DR\\+)"
ColName <- "CD8_Tem_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD45RA-.*(HLA-DR-CD103-|CD103-HLA-DR-)"
ColName <- "CD8_Tem_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

#__ CD8 Trm cells and activation markers
ColText <- "CD8\\+_CD62L-CD103\\+.*(HLA-DR\\+$)"
ColName <- "CD8_Trm_Activ1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD8\\+_CD62L-CD103\\+.*(HLA-DR-$)"
ColName <- "CD8_Trm_Activ0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

##________________________ NK T cells ____________________________________________________________________________########################
ColText <- "CD3\\+CD56\\+"
ColName <- "NKT_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3\\+CD56\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "NKT_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3\\+CD56\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "NKT_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3\\+CD56\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "NKT_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3\\+CD56\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "NKT_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

##________________________ NK cells _________________________________________________________________________________######################
ColText <- "CD3-CD56\\+"
ColName <- "NK_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3-CD56\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "NK_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3-CD56\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
```

```
ColName <- "NK_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3-CD56\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "NK_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3-CD56\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "NK_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

##______________________ B cells _________________________________________________________________________######################
ColText <- "(CD3-CD20\\+|CD3-CD19\\+)"
ColName <- "Bcells_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "(CD3-CD20\\+|CD3-CD19\\+).*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "Bcells_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "(CD3-CD20\\+|CD3-CD19\\+).*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "Bcells_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "(CD3-CD20\\+|CD3-CD19\\+).*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "Bcells_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "(CD3-CD20\\+|CD3-CD19\\+).*(PD-1-_LAG-3-TIM-3-)"
ColName <- "Bcells_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

##______________________ Myeloid cells __________________________________________________________________######################
ColText <- "CD3-CD33\\+"
ColName <- "Myeloid_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3-CD33\\+.*(PD-1\\+_LAG-3\\+TIM-3\\+)"
ColName <- "Myeloid_Inhib3_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3-CD33\\+.*(PD-1\\+_LAG-3\\+TIM-3-|PD-1\\+_LAG-3-TIM-3\\+|PD-1-_LAG-3\\+TIM-3\\+)"
ColName <- "Myeloid_Inhib2_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3-CD33\\+.*(PD-1\\+_LAG-3-TIM-3-|PD-1-_LAG-3\\+TIM-3-|PD-1-_LAG-3-TIM-3\\+)"
ColName <- "Myeloid_Inhib1_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

ColText <- "CD3-CD33\\+.*(PD-1-_LAG-3-TIM-3-)"
ColName <- "Myeloid_Inhib0_count"
phenotyping_analysis <- getRowSums(ColText, ColName)

##______ 4. Exporting Analysis table _______________________________________________________________________######################
# select for panels
phenotyping_analysis <- phenotyping_analysis[ panelID %in% c("1","2")]
phenotyping_analysis <- setorderv(phenotyping_analysis, c("panelID","panelSub"))
print(phenotyping_analysis)
write_csv(phenotyping_analysis, path = "ImmuNEO-15.2_P1-2_analysis.csv")

##______ 5. Merge all analysis files together ______________________________________________________________######################
#ImmuNEO37_analysis <- fread("ImmuNEO-37_P1-2_analysis.csv")
#ImmuNEO38_analysis <- fread("ImmuNEO-38_P1-2_analysis.csv")
#phenotyping_merged <- rbind(ImmuNEO37_analysis,ImmuNEO38_analysis, fill=T)
#phenotyping_merged

# load one analysis file to get the right order of columns because merging all files can result in change of column order. Use this to reset
ImmuNEO38_analysis <- fread("ImmuNEO-38_P1-2_analysis.csv")
colnametemplate <- colnames(ImmuNEO38_analysis)
colnametemplate
allfiles <- list.files(pattern = "analysis.csv")
phenotyping_merged <- data.table()
for(myfiles in allfiles){
  dt <- fread(myfiles)
  phenotyping_merged <- rbind(phenotyping_merged, dt, fill=T)
}
```

```
setcolorder(phenotyping_merged, colnametemplate)
phenotyping_merged

##______ 6. Calculate percentages, fuse Panles and calculate means _____________________________________________#######################
##____ Calculate percentages
allcols <- colnames(phenotyping_merged)[-1:-4]
for(mycol in allcols){
  newname <- gsub("_count", "_freq", mycol) # get the new column name with _freq instead of _count
  colIndex <- grep(mycol, colnames(phenotyping_merged)) # get the index of mycol
  neworder <- c(colnames(phenotyping_merged[, 1:colIndex]), newname) # get the list of columns BEFORE mycol + mycol and newname col
  phenotyping_merged[, eval(newname) := round(get(mycol) / singles * 100, digits = 3)]  # calculate percent for all rows of mycol
  setcolorder(phenotyping_merged, neworder) # sort the column order (all not named cols are ignored)
}
phenotyping_merged

##____ calculate means of count and percentage etc.
allcols2 <- colnames(phenotyping_merged)[-1:-4]
phenotyping_mean <- phenotyping_merged[, lapply(.SD, mean, na.rm=T), by=.(panelID, patient), .SDcols=allcols2]
phenotyping_mean[phenotyping_mean == "NaN"] <- NA

#___edit the table by hand (remove "wrong" values from Panel 2-2 for CD4 cells)
phenotyping_mean_edited <- fread("Phenotyping_allPatients_merged_means_edited.csv")
#phenotyping_mean_edited <- phenotyping_mean_edited[,lapply(.SD,as.numeric), by = allcols2]
phenotyping_mean_edited[phenotyping_mean_edited == ""] <- NA
phenotyping_mean_edited[phenotyping_mean_edited == "NA"] <- NA
phenotyping_mean_edited[phenotyping_mean_edited == " "] <- NA

##___ dense the data from all panels together in 1 row per patient
phenotyping_densPerPatient <- phenotyping_mean_edited[, lapply(.SD, max, na.rm=T), by=patient, .SDcols=allcols2]
phenotyping_densPerPatient[phenotyping_densPerPatient == "-Inf"] <- NA
setorder(phenotyping_densPerPatient, patient)

phenotyping_densPerPatient <- phenotyping_densPerPatient[, lapply(.SD, function(x){round(x,digit=3)}) ]

##____ calculate the number of cells per gramm tumor
tumorMasses <- fread("ImmuNEO_tumorMasses_cellNumbers.csv")
phenotyping_densPerPatient_quant <- merge(tumorMasses, phenotyping_densPerPatient)

selectedcols <- colnames(phenotyping_densPerPatient_quant)[grep("freq", colnames(phenotyping_densPerPatient_quant))]
for(mycol in selectedcols){
  newname <- gsub("_freq", "_quant", mycol) # get the new column name with _quant instead of _freq
  colIndex <- grep(mycol, colnames(phenotyping_densPerPatient_quant)) # get the index of mycol
  neworder <- c(colnames(phenotyping_densPerPatient_quant[, 1:colIndex]), newname) # get the list of columns BEFORE mycol + mycol and
newname col
  phenotyping_densPerPatient_quant[, eval(newname) := round(totalCells_digest*(get(mycol)/100)*(1/Weight_tumor), digits=0), by=patient]  #
calculate percent for all rows of mycol
  setcolorder(phenotyping_densPerPatient_quant, neworder) # sort the column order (all not named cols are ignored)
}
phenotyping_densPerPatient_quant

##______ 7. Plots _______________________________________________________________________________________#######################
# load data
phenotyping_densPerPatient_quant <- fread("Phenotyping_allPatients_merged_means_edited_densed_quant.csv")

# transverse the table
# ! always check first and last column name
phenotyping_densdPerPatient_long <- data.table(gather(phenotyping_densPerPatient_quant, celltype, value,
lymphocytes_count:Myeloid_Inhib0_quant))

# assign sorting criteria "freq" or "count"
tmp <- phenotyping_densdPerPatient_long[grep("quant", celltype)]
tmp[, cellgroup := gsub("\\d_quant$", "", celltype)]
tmp[, cellgroup := gsub("_quant$", "", cellgroup)]
tmp[, nMarker := as.numeric(gsub(".*_.*(\\d)_quant","\\1", celltype))]
tmp[, celltype := gsub("_quant$", "", celltype)]
tmp <- phenotyping_densdPerPatient_long[grep("freq", celltype)]
tmp[, cellgroup := gsub("\\d_freq$", "", celltype)]
tmp[, cellgroup := gsub("_freq$", "", cellgroup)]
tmp[, nMarker := as.numeric(gsub(".*_.*(\\d)_freq","\\1", celltype))]
tmp[, celltype := gsub("_freq$", "", celltype)]
tmp

##__________ plot 1 - Display all general cell populations ______________________________________######
d1 <- tmp[is.na(nMarker), ]
d1[, cellgroup := NULL]
```

```
d1.1 <- d1[!grep("_", celltype)]
setorder(d1.1, celltype, patient)
d1.1[, group := patient]
d1.1[, group := factor(group, levels=rev(d1.1[, unique(group)]))]

#setorder(d1.1, -patient)
#d1.1[, celltype := factor(celltype, levels=d1.1[, unique(celltype)])]

myorder <- c("lymphocytes","CD45", "CD3","CD4", "CD8","NKT","NK", "Bcells", "Myeloid")
d1.1[, celltype := factor(celltype, levels=myorder)]

ggplot(d1.1, aes(group, value)) +
 geom_bar(stat = "identity") +
 coord_flip() +
 facet_wrap(~celltype, scales = "free") +
 labs(title="Number of cells per gramm tumor", x ="Patient ID", y = "# cells/grammtumor") +
 scale_y_continuous(labels=comma) #transforms strange number into numeric values
 #theme(axis.text.x = element_text(angle = 300))

ggplot(d1.1, aes(group, value)) +
 geom_bar(stat = "identity") +
 coord_flip() +
 facet_wrap(~celltype, scales = "free") +
 labs(title="Number of cells per gramm tumor", x ="Patient ID", y = "# cells/grammtumor") +
 scale_y_sqrt(labels=comma) #transforms strange number into numeric values
#theme(axis.text.x = element_text(angle = 300))

d1.2 <- d1.1[, value := log2(value)]
ggplot(d1.2, aes(group, value)) +
 geom_bar(stat = "identity") +
 coord_flip() +
 facet_wrap(~celltype, scales = "free") +
 labs(title="Number of cells per gramm tumor", x ="Patient ID", y = "# cells/grammtumor in log") +
 scale_y_continuous(labels=comma)

##___________ plot 2 - Display all T cell subtypes regardless of marker expression ______________ ######
# create table with celltypes without a marker annotation and delete the cellgroup column
d2 <- tmp[is.na(nMarker), ]
d2[, cellgroup := NULL]

# only take those celltypes that contain a "_" which are all T cell subtypes e.g. CD4_Tcm, CD8_Teff
d2.1 <- d2[grep("_", celltype)]
setorder(d2.1, celltype, patient)
d2.1[, group := patient]
d2.1[, group := factor(group, levels=rev(d2.1[, unique(group)]))]

ggplot(d2.1, aes(group, value)) +
 geom_bar(stat = "identity") +
 coord_flip() +
 facet_wrap(~celltype, scales = "free", ncol =3)+
 labs(title="Number of cells per gramm tumor", x ="Patient ID", y = "# cells/gramm tumor") +
 scale_y_continuous(labels=comma)

ggplot(d2.1, aes(group, value)) +
 geom_bar(stat = "identity") +
 coord_flip() +
 facet_wrap(~celltype, scales = "free", ncol =3)+
 labs(title="Number of cells per gramm tumor", x ="Patient ID", y = "# cells/gramm tumor") +
 theme(legend.key.size = unit(0.2, "cm"),
       axis.text.x = element_text(size= 13),
       axis.text.y = element_text(size= 13),
       plot.title = element_blank(),
       axis.title.y = element_text(size=20),
       axis.title.x = element_text(size=20),
       strip.text.x = element_text(size = 15)) +
 scale_y_sqrt(n.breaks = 4,labels=comma)

d2.2 <- d2.1[, value := log2(value)]
ggplot(d2.2, aes(group, value)) +
 geom_bar(stat = "identity") +
 coord_flip() +
 facet_wrap(~celltype, scales = "free", ncol = 3) +
 labs(title="Number of cells per gramm tumor", x ="Patient ID", y = "# cells/gramm tumor in log") +
 scale_y_continuous(labels=comma)

##___________ plot 3 - Stack all T cell subtypes without markers _________________________________ #####
```

```
d3 <- tmp[is.na(nMarker), ]
d3[, cellgroup := NULL]

d3.1 <- d3[grep("_", celltype)]
d3.1 <- cSplit(d3.1, "celltype", "_", drop = F)
setnames(d3.1, c("celltype_1", "celltype_2"), c("cellgroup", "subtype"))
d3.1[, patient := as.character(patient)]
d3.1[, group := patient]
d3.1[, group := factor(group, levels=rev(d3.1[, unique(group)]))]

# select subsets
d3.3 <- d3.1[!patient %in% c("19-1","19-2","19-3","19-4", "17-2" )]
d3.4 <- d3.1[patient %in% c("19-1","19-2","19-3","19-4", "17-2" )]

d3.1 <- d3.1[!d3.1$patient %in% c("34", "14", "30", "15.2", "09"), ]
d3.3 <- d3.3[!d3.3$patient %in% c("34", "14", "30", "15.2", "09"), ]

d3.1.2 <- d3.1[!d3.1$celltype %in% c("CD4_Tregs", "CD4_Trm", "CD8_Trm"), ]

# for frequencies of CD4/CD8, generate from quantified data set
d3.5 <- d3.1.2
d3.5[, total_cells := sum(value,na.rm = TRUE), by =list(cellgroup, group)]
d3.5[, percent_subtype := round(value / total_cells * 100, digits = 3)]
d3.5 <- d3.5[!d3.5$patient %in% c("34", "14", "30", "15.2", "09"), ]
d3.5_wide <- dcast(d3.5, patient ~ celltype, value.var = "percent_subtype" )

# Frequencies/percentages
#Frequencies of singel alive cells
ggplot(d3.1, aes(group, value, fill=subtype)) +
 geom_bar(stat = "identity", position = "stack") +
 coord_flip() +
 facet_wrap(~cellgroup) + # scales = "free"
 scale_fill_brewer(palette = "Set2") +
 labs( x ="Patient ID", y = "Frequency of all single alive cells") +
 #scale_y_continuous(labels=comma)+
 theme(legend.key.size = unit(0.5, "cm"),
       axis.text.x = element_text(size= 15),
       axis.text.y = element_text(size= 17),
       plot.title = element_blank(),
       axis.title.y = element_text(size=20),
       axis.title.x = element_text(size=20),
       legend.text = element_text(size= 20),
       legend.title = element_text(size=20, face = "bold"),
       strip.text.x = element_text(size = 15, face = "bold"))

#Frequencies of CD4/CD8 cells
ggplot(d3.5[!cellgroup == "CD8",], aes(group, percent_subtype, fill=subtype)) +
 geom_bar(stat = "identity", position = "stack") +
 coord_flip() +
 facet_wrap(~cellgroup) + # scales = "free"
 scale_fill_brewer(palette = "Set2") +
 labs( x ="Patient ID", y = "Frequency of all CD4+ T cells") +
 #scale_y_continuous(labels=comma)+
 theme(legend.key.size = unit(0.5, "cm"),
       axis.text.x = element_text(size= 15),
       axis.text.y = element_text(size= 17),
       plot.title = element_blank(),
       axis.title.y = element_text(size=20),
       axis.title.x = element_text(size=20),
       legend.text = element_text(size= 20),
       legend.title = element_text(size=20, face = "bold"),
       strip.text.x = element_text(size = 15, face = "bold"))

# Quantified
#patients with few cells
ggplot(d3.3, aes(group, value/1000000, fill=subtype)) +
 geom_bar(stat = "identity", position = "stack") +
 coord_flip() +
 facet_wrap(~cellgroup) + # scales = "free"
 scale_fill_brewer(palette = "Set2") +
 labs( x ="Patient ID", y = "Number of cells/gramm tumor in Mio") +
 #scale_y_continuous(labels=comma)+
 theme(legend.key.size = unit(0.5, "cm"),
       axis.text.x = element_text(size= 15),
       axis.text.y = element_text(size= 17),
       plot.title = element_blank(),
      axis.title.y = element_text(size=20),
      axis.title.x = element_text(size=20),
      legend.text = element_text(size= 20),
      legend.title = element_text(size=20, face = "bold"),
      strip.text.x = element_text(size = 15, face = "bold"))

ggplot(d3.3, aes(x = factor(group), y= value/sum(value), fill=subtype)) +
  geom_bar(stat = "identity", position = "stack") +
  coord_flip() +
  facet_wrap(~cellgroup) + # scales = "free"
  scale_fill_brewer(palette = "Set2") +
  labs( x ="Patient ID", y = "Number of cells/gramm tumor in Mio") +
  scale_y_continuous(labels=percent)+
  theme(legend.key.size = unit(0.5, "cm"),
      axis.text.x = element_text(size= 15),
      axis.text.y = element_text(size= 17),
      plot.title = element_blank(),
      axis.title.y = element_text(size=20),
      axis.title.x = element_text(size=20),
      legend.text = element_text(size= 20),
      legend.title = element_text(size=20, face = "bold"),
      strip.text.x = element_text(size = 15, face = "bold"))

# Patients with lots of cells
ggplot(d3.4, aes(group, value/1000000, fill=subtype)) +
  geom_bar(stat = "identity", position = "stack") +
  coord_flip() +
  facet_wrap(~cellgroup) + # scales = "free"
  scale_fill_brewer(palette = "Set2") +
  labs( x ="Patient ID", y = "Number of cells/gramm tumor in Mio") +
  scale_y_continuous(labels=comma)+
  theme(legend.key.size = unit(0.5, "cm"),
      axis.text.x = element_text(size= 15),
      axis.text.y = element_text(size= 17),
      plot.title = element_blank(),
      axis.title.y = element_text(size=20),
      axis.title.x = element_text(size=20),
      legend.text = element_text(size= 20),
      legend.title = element_text(size=20, face = "bold"),
      strip.text.x = element_text(size = 15, face = "bold"))

##___________ plot 4 - Stack all inhib. and activ. marker for all cell populations ______________######
d4 <- tmp[!is.na(nMarker)]
d4[, nMarker := as.character(nMarker)]

#__ Step 1 select subgroups for mapping if needed
# EXCLUDES  cell groups or patients
d4[,unique(cellgroup)] # displays all possible cell groups
d4.1 <- d4[grep( "CD4_Activ|CD4_Inhib|CD8_Activ|CD8_Inhib", cellgroup)]
 #or/and
#only graps these cell groups
d4.1 <- d4[grep( "CD4|CD8", cellgroup)]
d4.1 <- d4[grep( "CD8_Teff", cellgroup)]
d4.1 <- d4.1[grep( "CD4_Activ|CD4_Inhib|CD8_Activ|CD8_Inhib", cellgroup)]
d4.1 <- d4[grep( "NKT|NK|Bcells|Myeloid", cellgroup)]

#__ Step 2 plot in different ways
#_______ A) split by patients, stack by celltype and colour by marker
ggplot(d4.1, aes(cellgroup, value, fill=nMarker)) +
  geom_bar(stat = "identity", position = "stack") +
  coord_flip() +
  facet_wrap(~patient, scales = "free_x")

#_______ B) split by celltype, colour by marker
setorder(d4.1, cellgroup, patient)
d4.1[, group := patient]
d4.1[, group := factor(group, levels=rev(d4.1[, unique(group)]))]
d4.1[, cellgroup := factor(cellgroup, levels=c("CD4_Activ","CD8_Activ", "CD4_Inhib", "CD8_Inhib")) ]

d4.1 <- d4.1[!patient %in% c("19-1","19-2","19-3","19-4", "17-2" )]
d4.1 <- d4.1[!d4.1$patient %in% c("34", "14", "30", "15.2", "09"), ]

# for frequencies of CD4/CD8, generate from quantified data set
d4.3 <- d4.1
d4.3[, total_cells := sum(value,na.rm = TRUE), by =list(cellgroup, group)]
d4.3[, percent_marker := round(value / total_cells * 100, digits = 3)]
```

```
d4.3 <- d4.3[!d4.3$patient %in% c("34", "14", "30", "15.2", "09"), ]
d4.3_wide <- dcast(d4.3, patient ~ celltype, value.var = "percent_marker" )

# Frequencies/percentages
#Frequencies of singel alive cells
ggplot(d4.1, aes(group, value, fill=nMarker)) +
  geom_bar(stat = "identity", position = "stack") +
  coord_flip() +
  facet_wrap(~cellgroup, scales = "free") + # scales = "free"
  scale_fill_manual("nMarker", values = c("0" = "grey70", "1" = "yellowgreen", "2" = "orange", "3" = "red")) +
  labs( x ="Patient ID", y = "Frequency of all single alive cells") +
  #scale_y_continuous(labels=comma)+
  theme(legend.key.size = unit(0.5, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 17),
        plot.title = element_blank(),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 20),
        legend.title = element_text(size=20, face = "bold"),
        strip.text.x = element_text(size = 15, face = "bold"))

#Frequencies of CD4/CD8 cells
ggplot(d4.3[!cellgroup %in% c("CD4_Activ","CD4_Inhib"),], aes(group, percent_marker, fill=nMarker)) +
  geom_bar(stat = "identity", position = "stack") +
  coord_flip() +
  facet_wrap(~cellgroup) + # scales = "free"
  scale_fill_manual("nMarker", values = c("0" = "grey70", "1" = "yellowgreen", "2" = "orange", "3" = "red")) +
  labs( x ="Patient ID", y = "Frequency of all CD8+ T cells [%]") +
  #scale_y_continuous(labels=comma)+
  theme(legend.key.size = unit(0.5, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 17),
        plot.title = element_blank(),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 20),
        legend.title = element_text(size=20, face = "bold"),
        strip.text.x = element_text(size = 15, face = "bold"))

# Quantified
#patients with few cells
ggplot(d4.1, aes(group, value/1000000, fill=nMarker)) +
  geom_bar(stat = "identity", position = "stack") +
  coord_flip() +
  facet_wrap(~cellgroup)+
  labs( x ="Patient ID", y = "Number of cells/gramm tumor in Mio") +
  #scale_y_continuous(labels=comma)+
  theme(legend.key.size = unit(0.5, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 17),
        plot.title = element_blank(),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 20),
        legend.title = element_text(size=20, face = "bold"),
        strip.text.x = element_text(size = 15, face = "bold")) + #changes facet text size
  scale_fill_manual("nMarker", values = c("0" = "grey70", "1" = "yellowgreen", "2" = "orange", "3" = "red"))

#patients with many cells
d4.2 <- d4.1[patient %in% c("19-1","19-2","19-3","19-4", "17-2")]
ggplot(d4.2, aes(group, value/1000000, fill=nMarker)) +
  geom_bar(stat = "identity", position = "stack") +
  coord_flip() +
  facet_wrap(~cellgroup)+
  labs( x ="Patient ID", y = "Number of cells/gramm tumor in Mio") +
  scale_y_continuous(labels=comma)+
  theme(legend.key.size = unit(0.5, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 17),
        plot.title = element_blank(),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 20),
        legend.title = element_text(size=20, face = "bold"),
        strip.text.x = element_text(size = 15, face = "bold")) + #changes facet text size
  scale_fill_manual("nMarker", values = c("0" = "grey70", "1" = "yellowgreen", "2" = "orange", "3" = "red"))

# dodge the nMarker = 0 next to others
dcol <- d4[!patient %in% c("19-1","19-2","19-3","19-4", "17-2" )]
dcol <- dcol[grep( "CD4_Activ|CD4_Inhib|CD8_Activ|CD8_Inhib", cellgroup)]
setorder(dcol, cellgroup, patient)
dcol[, group := patient]
dcol[, group := factor(group, levels=rev(dcol[, unique(group)]))]
ynames <-  dcol[, unique(group)]

d4.1[, group := as.numeric(group)]
barwidth = 0.4
ggplot() +
  geom_bar(data=d4.1[nMarker != 0], aes(group, value, fill=nMarker), stat = "identity", position = "stack", width=barwidth) +
  geom_bar(data=d4.1[nMarker == 0], aes(group+0.4, value, fill = nMarker), stat = "identity", position = "stack", width=barwidth) +
  coord_flip() +
  facet_wrap(~cellgroup, ncol = 2)+
  labs(title="Number of cells per gramm tumor", x ="Patient ID", y = "# cells/gramm tumor") +
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 17),
        plot.title = element_blank(),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 15),
        legend.title = element_text(size=15, face = "bold"),
        strip.text.x = element_text(size = 15),
        panel.background = element_rect(fill="white"),
        panel.grid.minor.x = element_line(size=3),
        panel.grid.major.x = element_line(colour = "grey"),
        plot.background = element_rect(fill="white")) + #changes facet text size
  scale_fill_manual("nMarker", values = c("0" = "grey", "1" = "green3", "2" = "orange", "3" = "red"))+
  scale_x_continuous(breaks = seq(1, 23, by = 1), labels = rev(ynames) ) +
  scale_y_sqrt(labels=c("0,5Mio", "2Mio" ,"4Mio", "6Mio", "14Mio"), breaks = c(500000,2000000, 4000000 , 6000000, 14000000))
#(labels=c("0,5Mio", "2Mio" ,"4Mio", "8Mio", "14Mio"), breaks = c(500000,2000000, 4000000 , 8000000, 14000000) # for low patients plot
# labels=c("1Mio", "10Mio" ,"50Mio", "100Mio", "150Mio", "200Mio"), breaks = c(1000000,10000000, 50000000 , 100000000, 150000000,
200000000) # for high patients plot

## __________ plot 5 - Heatmap all celltypes quant _________________________________________________#####
# prepare data
tmp <- phenotyping_densdPerPatient[grep("quant", celltype)]
tmp[, cellgroup := gsub("\\d_quant$", "", celltype)]
tmp[, cellgroup := gsub("_quant$", "", cellgroup)]
tmp[, nMarker := as.numeric(gsub(".*_.*(\\d)_quant","\\1", celltype))]
tmp[, celltype := gsub("_quant$", "", celltype)]

d1 <- tmp[is.na(nMarker), ]
d1[, cellgroup := NULL]

## Extract important columns and change namings
d1.heat <- d1[, Weight_tumor := NULL][,totalCells_digest := NULL][, nMarker := NULL][!d1$patient == "15-2",]
d1.heat <- dcast(d1.heat, patient ~ celltype , value.var = "value")
d1.heat[, patient :=gsub("-", "_T", patient)][, patient := gsub("^", "IN_", patient)]
setnames(d1.heat, "patient", "Tumor_ID")

## Extract columns for annotation file and merge with references
d1.heat.anno <- d1.heat[,1:2]
d1.heat.anno$Patient_ID <- d1.heat.anno$Tumor_ID
d1.heat.anno$Metastasis <- d1.heat.anno$Tumor_ID
d1.heat.anno[, Patient_ID := gsub("_T.$", "", Patient_ID)][, Metastasis := gsub("IN_.._", "", Metastasis)][,Metastasis := gsub("IN_..", "T1",
Metastasis)]

d1.heat.anno <- merge(d1.heat.anno, reference.master, by = "Patient_ID")
setnames(d1.heat.anno, "Master_ID", "Master_ID_group")
d1.heat.anno <- d1.heat.anno %>% unite(Master_ID, c(Master_ID_group, Metastasis), sep = "_", remove = FALSE)
d1.heat.anno <- merge(d1.heat.anno, reference.entity, by = "Master_ID")
d1.heat.anno[, Bcells := NULL]

## Transform heatmap table into a matrix
d1.heat <- data.frame(d1.heat)
rownames_heat <- d1.heat$Tumor_ID
d1.heat$Tumor_ID <- NULL
row.names(d1.heat) <- rownames_heat

d1.heat.matrix <- as.matrix(d1.heat)
d1.heat.matrix[d1.heat.matrix == 0] <- 1
```

```
#d1.heat.matrix <- d1.heat.matrix[, -20]
#d1.heat.matrix <- d1.heat.matrix[, -4]
#d1.heat.matrix <- d1.heat.matrix[, -2]

## Create several annotation data frames for heatmap
annotate_entity <- d1.heat.anno[, 3:7]
annotate_entity[, Master_ID_group := NULL][, Metastasis := NULL][, Tumor_entity := NULL]
annotate_entity <- data.frame(annotate_entity)
rownames_anno <- annotate_entity$Tumor_ID
row.names(annotate_entity) <- rownames_anno
annotate_entity$Tumor_ID <- NULL

# heatmap 1 all together
Phenotyping_heatmap<- pheatmap(d1.heat.matrix.log10, fontsize = 12, annotation_row = annotate_entity, scale = "none", annotation_names_row
= FALSE, cutree_rows = 3, cutree_cols = 2)

#heatmap 2 only main patients
pheatmap(d1.heat.matrix.log10.main, fontsize = 12, annotation_row = annotate_entity, scale = "none", annotation_names_row = FALSE,
cutree_rows = 2, cluster_cols = FALSE, clustering_distance_rows = "euclidean")

#heatmap 3 only suppl. patients
Phenotyping_heatmap<- pheatmap(d1.heat.matrix.log10.suppl, fontsize = 12, annotation_row = annotate_entity, scale = "none",
annotation_names_row = FALSE, cluster_cols = FALSE, cluster_rows = FALSE)

Phenotyping_heatmap

tmp <- t(scale(t(d1.heat))) # Z-score on each row = celltype-wise
tmp <- scale(d1.heat) # Z-score on each colum = Patients-wise
Phenotyping_heatmap<- pheatmap(tmp, annotation_row = annotate_entity, scale = "none", annotation_names_row = FALSE)

## _________ plot 6 - T cell marker ratios _______________________________________________#####
Tcell_marker_ratios <- fread("Analysis lists/Marker_CD8CD4_ratios_percent.csv")
Tcell_marker_ratios[ ,Marker := gsub(" marker", "", Marker)]
Tcell_marker_ratios[ ,Tumor_ID := gsub("ImmuNEO", "IN", Tumor_ID)]

ggplot(Tcell_marker_ratios, aes(Tumor_ID, Ratio, colour=Marker)) +
 geom_segment( aes(x=Tumor_ID, xend=Tumor_ID, y=1, yend=Ratio), color="grey")+
 geom_hline(yintercept=1, colour="black", lwd=2)+
 geom_point(size = 4)+
 scale_y_continuous(breaks = c(0,1,2,4,6), labels = c(0,1,2,4,6))+
 facet_wrap('Cell_type')+
 coord_flip()+
 labs(x ="Tumor ID", y = "Ratio of cells per gram tumor with marker/without marker") +
 theme(legend.key.size = unit(0.2, "cm"),
       axis.text.x = element_text(size= 20),
       axis.text.y = element_text(size= 20),
       plot.title = element_blank(),
       axis.title.y = element_text(size=20),
       axis.title.x = element_text(size=20),
       legend.text = element_text(size= 20),
       legend.title = element_text(size=20, face = "bold"),
       strip.text.x = element_text(size = 20),
       panel.grid.minor.x = element_blank(),
       panel.border = element_blank()) # for size of Facet Label

ggplot(Tcell_marker_ratios, aes(Tumor_ID, Value, colour=Marker)) +
 geom_segment( aes(x=Tumor_ID, xend=Tumor_ID, y=1, yend=Value), color="grey", linetype= 2)+
 geom_hline(yintercept=1, colour="darkgrey", lwd=2)+
 geom_point(size = 4)+
 scale_y_continuous(breaks = c(0,1,2,4,6), labels = c(0,1,2,4,6))+
 scale_x_discrete(limits = rev)+
 scale_color_manual(values = c("#00BFC4", "#F8766D"))+
 facet_wrap('Cell_type')+
 coord_flip()+
 labs(x ="Tumor ID", y = "Ratio of cells per gram tumor with marker/without marker") +
 theme_light() +
 theme(legend.key.size = unit(0.2, "cm"),
       axis.text.x = element_text(size= 20),
       axis.text.y = element_text(size= 20),
       plot.title = element_blank(),
       axis.title.y = element_text(size=20),
       axis.title.x = element_text(size=20),
       legend.text = element_text(size= 20),
       legend.title = element_text(size=20, face = "bold"),
       strip.text.x = element_text(size = 20),
       panel.grid.minor.x = element_blank()) # for size of Facet Label
```

```
## exclude samples with only few panels analyzed from min figure and only show CD8
Tcell_marker_ratios_main <- Tcell_marker_ratios[!Tcell_marker_ratios$Tumor_ID == c("IN-34", "IN-14", "IN-30", "IN-15.2", "IN-05"), ]
Tcell_marker_ratios_main <- Tcell_marker_ratios_main[Tcell_marker_ratios_main$Cell_type == "CD8", ]
Tcell_marker_ratios_main <- Tcell_marker_ratios_main[Tcell_marker_ratios_main$Cell_type == "CD4", ]

ggplot(Tcell_marker_ratios_main, aes(Tumor_ID, Ratio, colour=Marker)) +
  geom_segment( aes(x=Tumor_ID, xend=Tumor_ID, y=1, yend=Ratio), color="grey", linetype= 2)+
  geom_hline(yintercept=1, colour="darkgrey", lwd=2)+
  geom_point(size = 4)+
  #scale_y_continuous(breaks = c(0,0.5,1,2,4,6), labels = c(0,0.5,1,2,4,6))+
  scale_x_discrete(limits = rev)+
  scale_color_manual(values = c("#00BFC4", "#F8766D"))+
  facet_wrap('Marker', scales = "free")+
  coord_flip()+
  labs(x ="Tumor ID", y = "Ratio of cells per gram tumor with marker/without marker") +
  theme_light() +
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 20, angle = 45, vjust = 1, hjust=1),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 20),
        legend.title = element_text(size=20, face = "bold"),
        strip.text.x = element_text(size = 20),
        panel.grid.minor.x = element_blank()) # for size of Facet Label

##plot with frequencies/percentages

ggplot(Tcell_marker_ratios_main, aes(Tumor_ID, Percent_yes, colour=Marker)) +
  geom_segment( aes(x=Tumor_ID, xend=Tumor_ID, y=50, yend=Percent_yes), color="grey", linetype= 2)+
  geom_hline(yintercept=50, colour="darkgrey", lwd=2)+
  geom_point(size = 4)+
  scale_y_continuous(limits = c(0,100))+
  scale_x_discrete(limits = rev)+
  scale_color_manual(values = c("#00BFC4", "#F8766D"))+
  facet_wrap('Marker', scales = "free")+
  coord_flip()+
  labs(x ="Tumor ID", y = "Percent of total CD8 T cells with min. 1 marker expressed") +
  theme_light() +
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 20, angle = 45, vjust = 1, hjust=1),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=20),
        axis.title.x = element_text(size=20),
        legend.text = element_text(size= 20),
        legend.title = element_text(size=20, face = "bold"),
        strip.text.x = element_text(size = 20),
        panel.grid.minor.x = element_blank()) # for size of Facet Label

##______ 8. Create and save analysis tables _______________________________________________________#######################
d4 <- tmp[!is.na(nMarker)]
d4[, nMarker := as.character(nMarker)]

# EXCLUDES  cell groups or patients
d4[,unique(cellgroup)] # displays all possible cell groups
d4.1 <- d4[!patient %in% c("19-1","19-2","19-3","19-4", "17-2" )]
#or/and
#only graps these cell groups
d4.1 <- d4[grep( "CD4|CD8", cellgroup)]
d4.1 <- d4[grep( "CD8", cellgroup)]
d4.1 <- d4.1[grep( "CD4_Activ|CD4_Inhib|CD8_Activ|CD8_Inhib", cellgroup)]
d4.1 <- d4[grep( "CD4_Tregs_Activ|CD4_Tregs_Inhib", cellgroup)]
d4.1 <- d4[grep( "NKT|NK|Bcells|Myeloid", cellgroup)]

## ___________ Output the quants for CD8 for all different markerlevels __________________________________________######
List_CD8_marker_quant <- d4.1

# Add cellnumbers for markers>0 for inhib and activ
List_CD8_marker_quant_0 <- List_CD8_marker_quant[nMarker == 0]
List_CD8_marker_quant_0 <- List_CD8_marker_quant_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table
List_CD8_marker_quant_0 <- List_CD8_marker_quant_0[cellgroup %in% c ("CD8_Inhib", "CD8_Activ")]
List_CD8_marker_quant_0[is.na(List_CD8_marker_quant_0)] <- 0
```

```
List_CD8_marker_quant_123 <- List_CD8_marker_quant[nMarker > 0]
List_CD8_marker_quant_123 <- List_CD8_marker_quant_123[cellgroup %in% c ("CD8_Inhib", "CD8_Activ")]
List_CD8_marker_quant_123[is.na(List_CD8_marker_quant_123)] <- 0

List_CD8_marker_quant_123_combi <- List_CD8_marker_quant_123[, c := sum(value), by = .(patient, cellgroup)]
List_CD8_marker_quant_123_combi <- List_CD8_marker_quant_123_combi[nMarker ==1]

List_CD8_marker_quant_analysed <- rbind(List_CD8_marker_quant_0, List_CD8_marker_quant_123_combi )

List_CD8_marker_quant_analysed <- List_CD8_marker_quant_analysed[,value := NULL][,celltype := NULL]
colnames(List_CD8_marker_quant_analysed) <- gsub("^c$", "quant_cells_sum", colnames(List_CD8_marker_quant_analysed))
List_CD8_marker_quant_analysed[ ,nMarker := gsub("1", ">0", nMarker)]
setorder(List_CD8_marker_quant_analysed, patient)

## ___________ Output the quants for CD4 for all different markerlevels _______________________________________________#######
List_CD4_marker_quant <- d4.1

# Add cellnumbers for markers>0 for inhib and activ
List_CD4_marker_quant_0 <- List_CD4_marker_quant[nMarker == 0]
List_CD4_marker_quant_0 <- List_CD4_marker_quant_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table
List_CD4_marker_quant_0 <- List_CD4_marker_quant_0[cellgroup %in% c ("CD4_Inhib", "CD4_Activ")]
List_CD4_marker_quant_0[is.na(List_CD4_marker_quant_0)] <- 0

List_CD4_marker_quant_123 <- List_CD4_marker_quant[nMarker > 0]
List_CD4_marker_quant_123 <- List_CD4_marker_quant_123[cellgroup %in% c ("CD4_Inhib", "CD4_Activ")]
List_CD4_marker_quant_123[is.na(List_CD4_marker_quant_123)] <- 0

List_CD4_marker_quant_123_combi <- List_CD4_marker_quant_123[, c := sum(value), by = .(patient, cellgroup)]
List_CD4_marker_quant_123_combi <- List_CD4_marker_quant_123_combi[nMarker ==1]

List_CD4_marker_quant_analysed <- rbind(List_CD4_marker_quant_0, List_CD4_marker_quant_123_combi )

List_CD4_marker_quant_analysed <- List_CD4_marker_quant_analysed[,value := NULL][,celltype := NULL]
colnames(List_CD4_marker_quant_analysed) <- gsub("^c$", "quant_cells_sum", colnames(List_CD4_marker_quant_analysed))
List_CD4_marker_quant_analysed[ ,nMarker := gsub("1", ">0", nMarker)]
setorder(List_CD4_marker_quant_analysed, patient)

## ___________ Output the quants for Tregs for all different markerlevels ____________________________________________######
List_Tregs_marker_quant <- d4.1

# Add cellnumbers for markers>0 for inhib and activ
List_Tregs_marker_quant_0 <- List_Tregs_marker_quant[nMarker == 0]
List_Tregs_marker_quant_0 <- List_Tregs_marker_quant_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table
List_Tregs_marker_quant_0 <- List_Tregs_marker_quant_0[cellgroup %in% c ("CD4_Tregs_Inhib")]
List_Tregs_marker_quant_0[is.na(List_Tregs_marker_quant_0)] <- 0

List_Tregs_marker_quant_123 <- List_Tregs_marker_quant[nMarker > 0]
List_Tregs_marker_quant_123 <- List_Tregs_marker_quant_123[cellgroup %in% c ("CD4_Tregs_Inhib")]
List_Tregs_marker_quant_123[is.na(List_Tregs_marker_quant_123)] <- 0

List_Tregs_marker_quant_123_combi <- List_Tregs_marker_quant_123[, c := sum(value), by = .(patient, cellgroup)]
List_Tregs_marker_quant_123_combi <- List_Tregs_marker_quant_123_combi[nMarker ==1]

List_Tregs_marker_quant_analysed <- rbind(List_Tregs_marker_quant_0, List_Tregs_marker_quant_123_combi )

List_Tregs_marker_quant_analysed <- List_Tregs_marker_quant_analysed[,value := NULL][,celltype := NULL]
colnames(List_Tregs_marker_quant_analysed) <- gsub("^c$", "quant_cells_sum", colnames(List_Tregs_marker_quant_analysed))
List_Tregs_marker_quant_analysed[ ,nMarker := gsub("1", ">0", nMarker)]
setorder(List_Tregs_marker_quant_analysed, patient)

## ___________ Output the quants for CD8_Tcm for all different markerlevels _______________________________________________######
List_CD8_Tcm_marker_quant <- d4.1

# Add cellnumbers for markers>0 for inhib and activ
List_CD8_Tcm_marker_quant_0 <- List_CD8_Tcm_marker_quant[nMarker == 0]
List_CD8_Tcm_marker_quant_0 <- List_CD8_Tcm_marker_quant_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table
List_CD8_Tcm_marker_quant_0 <- List_CD8_Tcm_marker_quant_0[cellgroup %in% c ("CD8_Tcm_Inhib", "CD8_Tcm_Activ")]
List_CD8_Tcm_marker_quant_0[is.na(List_CD8_Tcm_marker_quant_0)] <- 0

List_CD8_Tcm_marker_quant_123 <- List_CD8_Tcm_marker_quant[nMarker > 0]
List_CD8_Tcm_marker_quant_123 <- List_CD8_Tcm_marker_quant_123[cellgroup %in% c ("CD8_Tcm_Inhib", "CD8_Tcm_Activ")]
List_CD8_Tcm_marker_quant_123[is.na(List_CD8_Tcm_marker_quant_123)] <- 0

## ___________ Output the quants for CD8_Teff for all different markerlevels _______________________________________________######
# quanitfied cell numbers
```

```
List_CD8_Teff_marker_quant <- d4.1

# Add quant cellnumbers for markers>0 for inhib and activ
List_CD8_Teff_marker_quant_0 <- List_CD8_Teff_marker_quant[nMarker == 0]
List_CD8_Teff_marker_quant_0 <- List_CD8_Teff_marker_quant_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table
List_CD8_Teff_marker_quant_0 <- List_CD8_Teff_marker_quant_0[cellgroup %in% c ("CD8_Teff_Inhib", "CD8_Teff_Activ")]
List_CD8_Teff_marker_quant_0[is.na(List_CD8_Teff_marker_quant_0)] <- 0

List_CD8_Teff_marker_quant_123 <- List_CD8_Teff_marker_quant[nMarker > 0]
List_CD8_Teff_marker_quant_123 <- List_CD8_Teff_marker_quant_123[cellgroup %in% c ("CD8_Teff_Inhib", "CD8_Teff_Activ")]
List_CD8_Teff_marker_quant_123[is.na(List_CD8_Teff_marker_quant_123)] <- 0

List_CD8_Teff_marker_quant_123_combi <- List_CD8_Teff_marker_quant_123[, c := sum(value), by = .(patient, cellgroup)]
List_CD8_Teff_marker_quant_123_combi <- List_CD8_Teff_marker_quant_123_combi[nMarker ==1]

List_CD8_Teff_marker_quant_analysed <- rbind(List_CD8_Teff_marker_quant_0, List_CD8_Teff_marker_quant_123_combi )

List_CD8_Teff_marker_quant_analysed <- List_CD8_Teff_marker_quant_analysed[,value := NULL][,celltype := NULL]
colnames(List_CD8_Teff_marker_quant_analysed) <- gsub("^c$", "quant_cells_sum", colnames(List_CD8_Teff_marker_quant_analysed))
List_CD8_Teff_marker_quant_analysed[ ,nMarker := gsub("1", ">0", nMarker)]
setorder(List_CD8_Teff_marker_quant_analysed, patient)

# Add quant cellnumbers for markers>0 for inhib and activ
List_CD8_Teff_marker_freq_0 <- List_CD8_Teff_marker_freq[nMarker == 0]
List_CD8_Teff_marker_freq_0 <- List_CD8_Teff_marker_freq_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table
List_CD8_Teff_marker_freq_0 <- List_CD8_Teff_marker_freq_0[cellgroup %in% c ("CD8_Teff_Inhib", "CD8_Teff_Activ")]
List_CD8_Teff_marker_freq_0[is.na(List_CD8_Teff_marker_freq_0)] <- 0

List_CD8_Teff_marker_freq_123 <- List_CD8_Teff_marker_freq[nMarker > 0]
List_CD8_Teff_marker_freq_123 <- List_CD8_Teff_marker_freq_123[cellgroup %in% c ("CD8_Teff_Inhib", "CD8_Teff_Activ")]
List_CD8_Teff_marker_freq_123[is.na(List_CD8_Teff_marker_freq_123)] <- 0


List_CD8_Teff_marker_freq_123_combi <- List_CD8_Teff_marker_freq_123[, c := sum(value), by = .(patient, cellgroup)]
List_CD8_Teff_marker_freq_123_combi <- List_CD8_Teff_marker_freq_123_combi[nMarker ==1]

List_CD8_Teff_marker_freq_analysed <- rbind(List_CD8_Teff_marker_freq_0, List_CD8_Teff_marker_freq_123_combi )

List_CD8_Teff_marker_freq_analysed <- List_CD8_Teff_marker_freq_analysed[,value := NULL][,celltype := NULL]
colnames(List_CD8_Teff_marker_freq_analysed) <- gsub("^c$", "quant_cells_sum", colnames(List_CD8_Teff_marker_freq_analysed))
List_CD8_Teff_marker_freq_analysed[ ,nMarker := gsub("1", ">0", nMarker)]
setorder(List_CD8_Teff_marker_freq_analysed, patient)

## ___________ Output the quants for CD8_Tem for all different markerlevels _______________________________________________######
d4 <- tmp[!is.na(nMarker)]
d4[, nMarker := as.character(nMarker)]
d4.1 <- d4[grep( "CD8_Tem", cellgroup)]

# quanitfied cell numbers
List_CD8_Tem_marker_quant <- d4.1

# Add quant cellnumbers for markers>0 for inhib and activ
List_CD8_Tem_marker_quant_0 <- List_CD8_Tem_marker_quant[nMarker == 0]
List_CD8_Tem_marker_quant_0 <- List_CD8_Tem_marker_quant_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table
List_CD8_Tem_marker_quant_0 <- List_CD8_Tem_marker_quant_0[cellgroup %in% c ("CD8_Tem_Inhib", "CD8_Tem_Activ")]
List_CD8_Tem_marker_quant_0[is.na(List_CD8_Tem_marker_quant_0)] <- 0

List_CD8_Tem_marker_quant_123 <- List_CD8_Tem_marker_quant[nMarker > 0]
List_CD8_Tem_marker_quant_123 <- List_CD8_Tem_marker_quant_123[cellgroup %in% c ("CD8_Tem_Inhib", "CD8_Tem_Activ")]
List_CD8_Tem_marker_quant_123[is.na(List_CD8_Tem_marker_quant_123)] <- 0

List_CD8_Tem_marker_quant_123_combi <- List_CD8_Tem_marker_quant_123[, c := sum(value), by = .(patient, cellgroup)]
List_CD8_Tem_marker_quant_123_combi <- List_CD8_Tem_marker_quant_123_combi[nMarker ==1]

List_CD8_Tem_marker_quant_analysed <- rbind(List_CD8_Tem_marker_quant_0, List_CD8_Tem_marker_quant_123_combi )

List_CD8_Tem_marker_quant_analysed <- List_CD8_Tem_marker_quant_analysed[,value := NULL][,celltype := NULL]
colnames(List_CD8_Tem_marker_quant_analysed) <- gsub("^c$", "quant_cells_sum", colnames(List_CD8_Tem_marker_quant_analysed))
List_CD8_Tem_marker_quant_analysed[ ,nMarker := gsub("1", ">0", nMarker)]
setorder(List_CD8_Tem_marker_quant_analysed, patient)

## ___________ Output the quants for CD8_Tn for all different markerlevels _______________________________________________######
d4 <- tmp[!is.na(nMarker)]
```

```
d4[, nMarker := as.character(nMarker)]
d4.1 <- d4[grep( "CD8_Tn", cellgroup)]

# quanitfied cell numbers
List_CD8_Tn_marker_quant <- d4.1

# Add quant cellnumbers for markers>0 for inhib and activ
List_CD8_Tn_marker_quant_0 <- List_CD8_Tn_marker_quant[nMarker == 0]
List_CD8_Tn_marker_quant_0 <- List_CD8_Tn_marker_quant_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table
List_CD8_Tn_marker_quant_0 <- List_CD8_Tn_marker_quant_0[cellgroup %in% c ("CD8_Tn_Inhib", "CD8_Tn_Activ")]
List_CD8_Tn_marker_quant_0[is.na(List_CD8_Tn_marker_quant_0)] <- 0

List_CD8_Tn_marker_quant_123 <- List_CD8_Tn_marker_quant[nMarker > 0]
List_CD8_Tn_marker_quant_123 <- List_CD8_Tn_marker_quant_123[cellgroup %in% c ("CD8_Tn_Inhib", "CD8_Tn_Activ")]
List_CD8_Tn_marker_quant_123[is.na(List_CD8_Tn_marker_quant_123)] <- 0

List_CD8_Tn_marker_quant_123_combi <- List_CD8_Tn_marker_quant_123[, c := sum(value), by = .(patient, cellgroup)]
List_CD8_Tn_marker_quant_123_combi <- List_CD8_Tn_marker_quant_123_combi[nMarker ==1]

List_CD8_Tn_marker_quant_analysed <- rbind(List_CD8_Tn_marker_quant_0, List_CD8_Tn_marker_quant_123_combi )

List_CD8_Tn_marker_quant_analysed <- List_CD8_Tn_marker_quant_analysed[,value := NULL][,celltype := NULL]
colnames(List_CD8_Tn_marker_quant_analysed) <- gsub("^c$", "quant_cells_sum", colnames(List_CD8_Tn_marker_quant_analysed))
List_CD8_Tn_marker_quant_analysed[ ,nMarker := gsub("1", ">0", nMarker)]
setorder(List_CD8_Tn_marker_quant_analysed, patient)

## ___________ Output the freq for all CD4 and CD8 T cell subtypes for all different markerlevels _______________________________________######
# load data
phenotyping_densPerPatient_quant <- fread("Phenotyping_allPatients_merged_means_edited_densed_quant.csv")
# transverse the table
# ! always check first and last column name
phenotyping_densdPerPatient_long <- data.table(gather(phenotyping_densPerPatient_quant, celltype, value, lymphocytes_count:Myeloid_Inhib0_quant))

#!!! CD4_Tcm_Activ2 is named wrongly and is in real CD8_Tcm_Activ2 --> rename (no data for CD4_Tcm_activ2 available, re-analysis for that cell type needed if wanted)
phenotyping_densdPerPatient_long[, celltype := gsub("CD4_Tcm_Activ2", "CD8_Tcm_Activ2", celltype)]

# assign sorting criteria "freq" or "count"
tmp <- phenotyping_densdPerPatient_long[grep("freq", celltype)]
tmp[, cellgroup := gsub("\\_freq$", "", celltype)]
tmp[, cellgroup := gsub('[0-3]+$', '', cellgroup)] # remove number of markers from Inhib and Activ (! CD3 will also be deleted to CD)
tmp[, cellgroup := gsub('^CD$', 'CD3', cellgroup)] # rename to CD3, because 3 was deleted in step before
#tmp[, cellgroup := gsub("_freq$", "", cellgroup)]
tmp[, nMarker := as.numeric(gsub(".*_.*(\\d)_freq","\\1", celltype))]
tmp[, celltype := gsub("_freq$", "", celltype)]

d4 <- tmp[!is.na(nMarker)]
d4[, nMarker := as.character(nMarker)]

# Add quant cellnumbers for markers>0 for inhib and activ
List_all_marker_freq_0 <- d4[nMarker == 0]
List_all_marker_freq_0 <- List_all_marker_freq_0 %>% mutate( c = value) #duplcate the "value" column and paste it to the end of the table, to later fuse with 123 value
#List_all_marker_freq_0 <- List_all_marker_freq_0[cellgroup %in% c ("CD8_Tn_Inhib", "CD8_Tn_Activ")]
List_all_marker_freq_0[is.na(List_all_marker_freq_0)] <- 0

List_all_marker_freq_123 <- d4[nMarker > 0]
#List_all_marker_freq_123 <- List_all_marker_freq_123[cellgroup %in% c ("CD8_Tn_Inhib", "CD8_Tn_Activ")]
List_all_marker_freq_123[is.na(List_all_marker_freq_123)] <- 0

List_all_marker_freq_123_combi <- List_all_marker_freq_123[, c := sum(value, na.rm = TRUE), by = .(patient, cellgroup)]
List_all_marker_freq_123_combi <- List_all_marker_freq_123_combi[nMarker ==1]

List_all_marker_freq_analysed <- rbind(List_all_marker_freq_0, List_all_marker_freq_123_combi )

List_all_marker_freq_analysed <- List_all_marker_freq_analysed[,value := NULL][,celltype := NULL]
colnames(List_all_marker_freq_analysed) <- gsub("^c$", "freq_cells_sum", colnames(List_all_marker_freq_analysed))
List_all_marker_freq_analysed[ ,nMarker := gsub("1", ">0", nMarker)]
setorder(List_all_marker_freq_analysed, patient)
List_all_marker_freq_analysed$cellgroup_marker <- paste(List_all_marker_freq_analysed$cellgroup, List_all_marker_freq_analysed$nMarker, sep = "_")

List_all_marker_freq_analysed_wide <- dcast(List_all_marker_freq_analysed, patient ~ cellgroup_marker, value.var = "freq_cells_sum" )
```

6.9.7 Integration of data and statistical analysis

```
library(data.table)
library(csv)
library(readxl)
library(writexl)
library(tidyr)
library(splitstackshape)
library(readr)
library(ggplot2)
library(dplyr)
library(scales)
library(xlsx)
library(ggpubr)
library(ggrepel)
library(corrplot)
library(survival)
library(mvtnorm)
library(ROCit)
library(tidyverse)
library(forestplot)
library(naniar)

## _______ set wd and load data ________________________________________#######
Integration_table <- fread("Table_Integration_V17_new_neoantigens.csv")

## ______ tidy data _____________________________________________________######
Integration_table[Integration_table == "x"] <- NA
Integration_table[Integration_table == "Ö"] <- "x"
#Integration_table <- Integration_table[, V77 := NULL][, V78 := NULL]
Integration_table <- Integration_table[1:42]

Integration_table$Phenotyping <- NULL
Integration_table$Sort <- NULL

colnames(Integration_table)
col_names <- colnames(Integration_table)
#changeCols <- c("Tumor mass [g]","Tumor mass for digest","Tumor mass for MS", "Cells after digest in Mio", "Cells per gramm tumor","cultivated
TILs in Mio","freq_CD3",
"quant_CD3","freq_CD8","quant_CD8","CD8_Inhib_no","CD8_Activ_no","CD8_Inhib_yes","CD8_Activ_yes","Ratio_Inhib_CD8","Ratio_Activ_CD8","f
req_CD8_Tcm","quant_CD8_Tcm","CD8_Tcm_Inhib_no","CD8_Tcm_Activ_no","CD8_Tcm_Inhib_yes","CD8_Tcm_Activ_yes","Ratio_Inhib_CD8_Tcm
","Ratio_Activ_CD8_Tcm","CD8_Tn_freq","CD8_Tn_quant","CD8_Teff_freq","CD8_Teff_quant","CD8_Tem_freq","CD8_Tem_quant","freq_CD4","qu
ant_CD4","CD4_Inhib_no","CD4_Activ_no","CD4_Inhib_yes","CD4_Activ_yes","Ratio_Inhib_CD4","Ratio_Activ_CD4","freq_Tregs","quant_Tregs","Tr
egs_Inhib_no","Tregs_Inhib_yes","Ratio_Inhib_Tregs","Ratio_quant_CD8_CD4","Ratio_marker_CD8","Ratio_marker_CD4","Ratio_Inhib_CD8_CD4","
Ratio_Activ_CD8_CD4","mutational_load","wt_peptidome_5FDR","wt_peptidome_1FDR","quant_wt_peptides_5FDR","quant_wt_peptides_1FDR","
peptides_MS","peptides_prediction","Immunetherapy","Response","Survival","Survival_ID_months","Survival_MD_months",
"Survival_MD_1year","Survival_MD_2years","Survival_MD_meanyears","Survival_MASTER_months" )
changeCols_2 <- col_names[! col_names %in% c("Sample_ID","Patient_ID","MASTER_ID","Partner_site","Tumor_entity","Metastatic_site",
"Initial_Diagnosis", "Phenotyping")]
Integration_table_new <- Integration_table[,(changeCols_2):= lapply(.SD, as.numeric), .SDcols = changeCols_2] # change to numeric values

## ______ Subset data __________________________________________________######
# Combine neoantigens from several metastasis per Patient
#Integration_table_new[, c("peptides_MS")][is.na(Integration_table_new[, c("peptides_MS")])] <- 0 #replace NAs by 0 to get the sums only for that
column
#Integration_table_new[,peptides_MS:=sum(peptides_MS),by = Patient_ID] #Update the column peptides_MS by the sum by Patient_ID

# Select one from multiple metastasis
#Integration_table_woLN <- Integration_table_new[!Metastatic_site %in% c("LN","LN colon","retroperitoneal LN" )]
Integration_table_uniquePatients <- Integration_table_new[!MASTER_ID %in% c("GXL1B7_T2", "64EMZ9_T1", "Q1PB42_T1", "Q1PB42_T3",
"1MULDR_T1","1MULDR_T2","1MULDR_T3","NVDER5_T1", "LFNUX6_T2", "ATE46U_T1")]

Integration_table_uniquePatients_ImmuTh <- Integration_table_uniquePatients[Immunotherapy %in% c(1)]

# check for normality of specific parameters
# via density plot
ggdensity(Integration_table_uniquePatients$peptides_MS_perTumor,
        main = "Density plot of factor X",
        xlab = "Factor X")
# via Shapiro-Wilk’s test
shapiro.test(Integration_table_uniquePatients$peptides_MS_perTumor) # if bigger than 0.05 than the data is not significantly different from normal
distribution
# >0.05 normally distributed, <0.05 NOT normally distributed

## ______ Group correlations ___________________________________________######
```

```
dt <- Integration_table_new
dt <- Integration_table_uniquePatients
dt <- Integration_table_woLN
dt <- Integration_table_uniquePatients_ImmuTh

# ______________ single correlation  __________________________________________######
wilcox.test(Metastasis_Thrombocytes ~ peptides_MS_reactive_class, data = dt)$p.value
t.test(Metastasis_Thrombocytes ~ peptides_MS_reactive_class, data = dt)$p.value
kruskal.test(wt_peptidome_1FDR ~ Response_type, data = dt)$p.value

t.test(wt_peptidome_1FDR ~ Survival, data = dt)
t.test(quant_wt_peptides_5FDR ~ Survival, data = dt)
t.test(quant_wt_peptides_1FDR ~ Survival, data = dt)

# ______________ T test on all parameters ____________________________________#####
testdt <- copy(dt)
colnames(testdt) <- gsub(" ", "", colnames(testdt))
colnames(testdt) <- gsub("\\[.*\\]", "", colnames(testdt))
newchangeCols <- gsub(" ", "", changeCols_2)
newchangeCols <- gsub("\\[.*\\]", "", newchangeCols)

#decide for refcol
refcol <- "Response"
refcol <- "Response_post"
refcol <- "Survival"
refcol <- "Survival_MD_1year"
refcol <- "Survival_MD_2years"
refcol <- "Survival_MD_meanyears"

refcol <- "Survival_MASTER_1year"
refcol <- "Survival_MASTER_2years"
refcol <- "Survival_MASTER_meanyears"

refcol <- "peptides_MS_reactive_class"

#check if some testcols need to be excluded manually
testcols <- newchangeCols[!newchangeCols %in% refcol]

# leave out those cols where error accures because too little oberservations
#use this for survivial general
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("HLAI_per_tumor")]
#use this for MD 1_year correlations
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("Ratio_Inhib_CD8_Tn", "Ratio_Activ_CD8_Tn",
"Ratio_Activ_CD8_Tem", "Tregs_Inhib_no", "Tregs_Inhib_yes", "Ratio_Inhib_Tregs", "HLAI_per_tumor","CD3_percent_IHC", "TCR_diversity",
"Response_prior", "Response_prior_type")]
#use this for MD 2_year and MD mean_year and MASTER 1_year and MASTER mean_years correlations
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("HLAI_per_tumor","CD3_percent_IHC", "TCR_diversity",
"Response_prior", "Response_prior_type")]
#use this for  MASTER 2_years correlations
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("HLAI_per_tumor","CD3_percent_IHC")]

#use for Immunot. patients and Response & Response_post
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c( "Ratio_Activ_CD8_Tn","HLAI_per_tumor","CD3_percent_IHC",
"TCR_diversity", "Response_prior", "Response_prior_type")]

#use all patients and correl. to reactive neoantigens
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("Reactive_neoantigens_perPatient_yesno","Response")]
ttestdt <- data.table()
for (col in testcols){
 myname <- paste0(col, "~", refcol) # set the output name
 print(myname)
 nobs <- nrow(na.omit(testdt[, col, with=F])) # count the numbe r of observations (we need at least 22)
 print(nobs)
 only01 <- all(testdt[, get(col)] %in% c(0,1)) # check if the values are only 0 and 1
 # only run the t-test, if a) observations more than 22 and b) not only 0 and 1
 # else set pval to NA
 if(nobs > 1 & !only01){
  mytestformula <- as.formula(myname) # convert the name to formula for ttest
  pval <- t.test(mytestformula, data = testdt)$p.value # run t-test and extract pvalue
  mydt <- data.table(name = myname, pval) # generate a single row data.table with the result + name
  ttestdt <- rbind(ttestdt, mydt) # bind the small dt to the final large output data.table
 }
 else {
  mydt <- data.table(name = myname, pval=NA) # set pval to NA
  ttestdt <- rbind(ttestdt, mydt) # also bind to large final output table
 }
```

```
}
ttestdt[, padj := p.adjust(pval, method="BH")] # do pvalue adjustment for multiple testing

ttestdt_2 <- cSplit(ttestdt, "name", "~")
colnames(ttestdt_2) <- gsub("name_1", "Parameter", colnames(ttestdt_2))
colnames(ttestdt_2) <- gsub("name_2", "Reference", colnames(ttestdt_2))
setcolorder(ttestdt_2, c("Parameter", "Reference", "pval", "padj"))

# check values
ttestdt_2[pval < 0.05]
ttestdt_2[padj < 0.05]

## plot significant correlations
#create matrix
ttestdt_2_filtered <- as.data.frame(ttestdt_2[ttestdt_2$Parameter %in%
c("freq_CD3","freq_CD8","CD8_Inhib_no_freq_parent","CD8_Inhib_yes_freq_parent","CD8_Teff_freq_total","CD8_Teff_Inhib_no_freq_parent","CD
8_Teff_Inhib_yes_freq_parent","CD8_Tem_freq_total","CD8_Tem_Inhib_no_freq_parent",
"CD8_Tem_Inhib_yes_freq_parent","wt_peptidome_1FDR")])
ttestdt_2_filtered$pinverse <- as.numeric(1- ttestdt_2_filtered$pval)
ttestdt_2_filtered$padj <- NULL
ttestdt_2_filtered$pval <- NULL

ttestdt_2_filtered <- ttestdt_2_filtered %>%
  replace_with_na(replace = list(pinverse = c(0.749415, 0.7339711)))

ttestdt_2_filtered[ttestdt_2_filtered == 0.749415] <- NA

ttestdt_2_filtered <- reshape(ttestdt_2_filtered, idvar = "Reference", timevar = "Parameter", direction = "wide")
ttestdt_2_filtered$Reference <- NULL
ttestdt_2_filtered <- as.matrix(ttestdt_2_filtered)
rownames(ttestdt_2_filtered) <- c("peptides_MS_reactive_class")

corrplot(ttestdt_2_filtered,
        #method = 'number',
        tl.col="black",
        is.corr = FALSE,
        col = col2(100),
        tl.cex = 0.75, #0.75
        cl.lim = c(0.73,1),
        diag = TRUE,
        cl.cex = 0.75,cl.ratio = 0.1, tl.srt = 45) # (cl.cex = size of legend text, cl.ratio = distance of test from legend)
        #col.lim = c(0,1))
        #p.mat = corDT_p_mat, sig.level = 0.05, insig = "blank", outline = FALSE)

# _____________ Mann Whitney U test on all parameters __________________#####
testdt <- copy(dt)
colnames(testdt) <- gsub(" ", "", colnames(testdt))
colnames(testdt) <- gsub("\\[.*\\]", "", colnames(testdt))
newchangeCols <- gsub(" ", "", changeCols_2)
newchangeCols <- gsub("\\[.*\\]", "", newchangeCols)

#decide for refcol
refcol <- "Response"
refcol <- "Response_post"

refcol <- "Shared_RNAall_mut_12"

refcol <- "wt_peptides_shared"
refcol <- "peptides_MS_perPatient_yesno"
refcol <- "Reactive_neoantigens_perPatient_yesno"

#refcol <- "Survival"
#refcol <- "Survival_MD_1year"
#refcol <- "Survival_MD_2years"
#refcol <- "Survival_MD_meanyears"
#refcol <- "Survival_MASTER_1year"
#refcol <- "Survival_MASTER_2years"
#refcol <- "Survival_MASTER_meanyears"

#check if some testcols need to be excluded manually: Mutation_IL10RBDT, Reactive_neoantigens_perPatient_yesno
testcols <- newchangeCols[!newchangeCols %in% refcol]

# leave out those cols where error accures because too little oberservations
# use this for survival, MD 2_years, MD mean_years, Master 1_year, Master 2_years correlations
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("HLAI_per_tumor")]
# use this for MD 1year, and for Immunot. patients Response and Response_post
```

```
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("HLAI_per_tumor", "CD3_percent_IHC", "TCR_diversity")]
# use this for peptides_MS_perPatient_yesno
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("HLAI_per_tumor", "CD3_percent_IHC", "TCR_diversity",
"Reactive_neoantigens_perTumor", "Reactive_neoantigens_perPatient", "Reactive_neoantigens_perPatient_yesno" )]

utestdt <- data.table()
for (col in testcols){
  myname <- paste0(col, "~", refcol) # set the output name
  print(myname)
  nobs <- nrow(na.omit(testdt[, col, with=F])) # count the numbe r of observations (we need at least 22)
  print(nobs)
  only01 <- all(testdt[, get(col)] %in% c(0,1)) # check if the values are only 0 and 1
  # only run the t-test, if a) observations more than 22 and b) not only 0 and 1
  # else set pval to NA
  if(nobs > 1 & !only01){
    mytestformula <- as.formula(myname) # convert the name to formula for utest
    pval <- wilcox.test(mytestformula, data = testdt)$p.value # run t-test and extract pvalue
    mydt <- data.table(name = myname, pval) # generate a single row data.table with the result + name
    utestdt <- rbind(utestdt, mydt) # bind the small dt to the final large output data.table
  }
  else {
    mydt <- data.table(name = myname, pval=NA) # set pval to NA
    utestdt <- rbind(utestdt, mydt) # also bind to large final output table
  }
}

utestdt[, padj := p.adjust(pval, method="BH")] # do pvalue adjustment for multiple testing

utestdt <- cSplit(utestdt, "name", "~")
colnames(utestdt) <- gsub("name_1", "Parameter", colnames(utestdt))
colnames(utestdt) <- gsub("name_2", "Reference", colnames(utestdt))
setcolorder(utestdt, c("Parameter", "Reference", "pval", "padj"))

# check values
utestdt[pval < 0.05]
utestdt_2[padj < 0.05]

# _____________ combine both tables ________________________________________######
### T-Tests
Ttests_merged <- fread("Ttest_table_survival.csv")
Ttests_merged$Survival_padj <- NULL
Ttests_merged$Survival_pval <- NULL

allfiles <- list.files(pattern = ".csv")
for(myfiles in allfiles){
  dt <- fread(myfiles)
  Ttests_merged <- merge(Ttests_merged, dt, by = "Parameter", all = TRUE)
}

Ttests_merged_pval <- Ttests_merged
Ttests_merged_pval <- Ttests_merged_pval[, grep("padj", names(Ttests_merged_pval)) := NULL]
colnames(Ttests_merged_pval) <- gsub("^", "Ttest_", colnames(Ttests_merged_pval))
colnames(Ttests_merged_pval) <- gsub("Ttest_Parameter", "Parameter", colnames(Ttests_merged_pval))

### U-Tests
Utests_merged <- fread("Utest_table_responseIT.csv")
Utests_merged$Response_padj <- NULL
Utests_merged$Response_pval <- NULL

allfiles <- list.files(pattern = ".csv")
for(myfiles in allfiles){
  dt <- fread(myfiles)
  Utests_merged <- merge(Utests_merged, dt, by = "Parameter", all = TRUE)
}

Utests_merged_pval <- Utests_merged
Utests_merged_pval <- Utests_merged_pval[, grep("padj", names(Utests_merged_pval)) := NULL]
colnames(Utests_merged_pval) <- gsub("^", "Utest_", colnames(Utests_merged_pval))
colnames(Utests_merged_pval) <- gsub("Utest_Parameter", "Parameter", colnames(Utests_merged_pval))

### merge both data tables with T and U Test
Merged_T_U_test <- merge(Ttests_merged_pval, Utests_merged_pval, by = "Parameter", all = TRUE)

# _____________ Kruskal-Wallis rank sum H test on all parameters ________#####
testdt <- copy(dt)
colnames(testdt) <- gsub(" ", "", colnames(testdt))
colnames(testdt) <- gsub("\\[.*\\]", "", colnames(testdt))
newchangeCols <- gsub(" ", "", changeCols_2)
newchangeCols <- gsub("\\[.*\\]", "", newchangeCols)

#decide for refcol
refcol <- "Response_type"
refcol <- "Response_post_type"
refcol <- "Response_prior_type"

#check if some testcols need to be excluded manually
testcols <- newchangeCols[!newchangeCols %in% refcol]
# leave out those cols where error accures because too little oberservations
# use this for response type prior
testcols <- newchangeCols[!newchangeCols %in% refcol & !newchangeCols %in% c("HLAI_per_tumor", "CD3_percent_IHC", "TCR_diversity", "Myeloid_freq", "Myeloid_quant", "TMB_probe")]

htestdt <- data.table()
for (col in testcols){
  myname <- paste0(col, "~", refcol) # set the output name
  print(myname)
  nobs <- nrow(na.omit(testdt[, col, with=F])) # count the numbe r of observations (we need at least 22)
  print(nobs)
  only01 <- all(testdt[, get(col)] %in% c(0,1)) # check if the values are only 0 and 1
  # only run the t-test, if a) observations more than 22 and b) not only 0 and 1
  # else set pval to NA
  if(nobs > 1 & !only01){
    mytestformula <- as.formula(myname) # convert the name to formula for utest
    pval <- kruskal.test(mytestformula, data = testdt)$p.value # run t-test and extract pvalue
    mydt <- data.table(name = myname, pval) # generate a single row data.table with the result + name
    htestdt <- rbind(htestdt, mydt) # bind the small dt to the final large output data.table
  }
  else {
    mydt <- data.table(name = myname, pval=NA) # set pval to NA
    htestdt <- rbind(htestdt, mydt) # also bind to large final output table
  }
}

htestdt[, padj := p.adjust(pval, method="BH")] # do pvalue adjustment for multiple testing
htestdt <- cSplit(htestdt, "name", "~")
colnames(htestdt) <- gsub("name_1", "Parameter", colnames(htestdt))
colnames(htestdt) <- gsub("name_2", "Reference", colnames(htestdt))
setcolorder(htestdt, c("Parameter", "Reference", "pval", "padj"))

# check values
htestdt[pval < 0.05]
#utestdt_2[padj < 0.05]

## check significant results by paired U-test
pairwise.wilcox.test(dt$Mutations_RNAediting, dt$Response_type, p.adjust.method = "BH")
dt.sub<-dt[which(dt$Response_type!='2'),]
wilcox.test(peptides_MS_perPatient ~ Response_type, data = dt.sub)

##_________ Linear regression correlation ___________________________________#####
subtable <- Integration_table_uniquePatients[Patient_ID %in% c("ImmuNEO-05", "ImmuNEO-08", "ImmuNEO-15", "ImmuNEO-19", "ImmuNEO-17","ImmuNEO-11","ImmuNEO-28","ImmuNEO-25", "ImmuNEO-04", "ImmuNEO-20", "ImmuNEO-37", "ImmuNEO-23", "ImmuNEO-32", "ImmuNEO-38", "ImmuNEO-18", "ImmuNEO-35", "ImmuNEO-24")]

subtable <- Integration_table_uniquePatients[, changeCols_2, with = FALSE]
subtable <- subtable[,10:162]
subtable$wt_peptides_shared <- NULL
subtable$peptides_MS_perPatient_yesno <- NULL
subtable$Reactive_neoantigens_perPatient_yesno <- NULL
subtable <- subtable[,!90:92]
subtable$mutational_load_DNA <- NULL
subtable$mutational_load_RNA <- NULL
subtable$Mutations_RNAediting_filtered <- NULL

subtable <- as.data.frame(subtable)

corDT <- data.table(cor(subtable, use = "pairwise.complete.obs", method = "spearman"), keep.rownames = T)
#corDT <- data.table(cor(subtable, use = "pairwise.complete.obs", method = "pearson"), keep.rownames = T)
corDT

# extract p-values
cor.test(subtable$peptides_MS_perTumor, subtable$wt_peptidome_1FDR, method = "spearman" )[["p.value"]]
cor(subtable$peptides_MS_perTumor, subtable$mutational_load, method = "spearman" )[["p.value"]]
```

```
cor.test.p <- function(x){
  FUN <- function(x, y) cor.test(x, y, method = "spearman")[["p.value"]]
  z <- outer(
    colnames(x),
    colnames(x),
    Vectorize(function(i,j) FUN(x[,i], x[,j]))
  )
  dimnames(z) <- list(colnames(x), colnames(x))
  z
}

corDT_p <- data.table(cor.test.p(subtable), keep.rownames = T)

# single correlations
cor(Integration_table_uniquePatients$mutational_load, Integration_table_uniquePatients$peptides_MS, use = "complete.obs", method =
"pearson")
cor(Integration_table_uniquePatients$mutational_load, Integration_table_uniquePatients$peptides_MS, use = "complete.obs", method =
"spearman")
cor(Integration_table_uniquePatients$mutational_load, Integration_table_uniquePatients$peptides_MS, use = "complete.obs", method = "kendall")
cor(Integration_table_uniquePatients[, .(quant_CD3, quant_CD8, quant_CD8_Tcm)], use = "complete.obs")

##_______ plots ____________________________________________________________#####
# _____________ box plots correlations ___________________________________#####
ggplot(Integration_table_new, aes(Mutation_IL10RBDT, peptides_MS_perTumor, group = Mutation_IL10RBDT))+
  geom_boxplot()+
  scale_x_continuous(labels = c("not identified", "identified"), breaks = c(0 , 1))+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=.5) +
  #scale_y_continuous(labels = comma)+
  #labs(x ="Response to ICB after admission", y = "Predicted 9mer neoantigen candidates RNA editing") + #\n
  #geom_text_repel(data = Integration_table_uniquePatients_ImmuTh, aes(label = Patient_ID))+
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 20),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25))

## boxplot for response to IT comparisons
ggplot(Integration_table_uniquePatients, aes(Response, mutational_load, group = Response))+
  geom_boxplot()+
  scale_x_continuous(labels = c("no", "yes"), breaks = c(0 , 1))+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=.5) +
  #scale_y_continuous(labels = comma)+
  #labs(x ="Response to ICB general", y = "Number of total mutations") + #\n
  #geom_text_repel(data = Integration_table_uniquePatients, aes(label = Patient_ID))+
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 20),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25))

## boxplot for response TYPE to IT comparison
ggplot(Integration_table_uniquePatients, aes(Response_type, wt_peptidome_1FDR, group = Response_type))+
  geom_boxplot()+
  scale_x_continuous(labels = c("no", "mixed", "good"), breaks = c(0 , 1, 2))+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=.5) +
  #scale_y_continuous(labels = comma)+
  #labs(x ="Response to ICB general", y = "Wt peptidome 1% FDR") + #\n
  #geom_text_repel(data = Integration_table_uniquePatients_ImmuTh, aes(label = Patient_ID))+
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 20),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25))

## boxplot for reactive neoantigen correlation revision
ggplot(Integration_table_uniquePatients, aes(peptides_MS_reactive_class, wt_peptidome_1FDR, group = peptides_MS_reactive_class))+
  geom_boxplot()+
  scale_x_continuous(labels = c("no", "yes"), breaks = c(0 , 1))+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=.5) +
  #scale_y_continuous(labels = comma)+
  #labs(x ="Reactive neoantigen candidates found", y = "Frequency CD3 cells") + #\n
  #geom_text_repel(data = Integration_table_uniquePatients, aes(label = Patient_ID))+
  theme(legend.key.size = unit(0.2, "cm"),
        axis.text.x = element_text(size= 20),
        axis.text.y = element_text(size= 20),
        plot.title = element_blank(),
        axis.title.y = element_text(size=25),
        axis.title.x = element_text(size=25))

# _____________ scatter plots _____________________________________________#####
ggscatter(Integration_table_uniquePatients, x = "Metastasis_Thrombocytes", y = "peptides_MS_reactive",
          color = "black", shape = 21, size = 3, # Points color, shape and size
          add = "reg.line",  # Add regressin line
          add.params = list(color = "blue", fill = "lightgray"), # Customize reg. line
          conf.int = TRUE, # Add confidence interval
          cor.coef = TRUE, # Add correlation coefficient. see ?stat_cor
          cor.coeff.args = list(method = "spearman", label.x = 0, label.sep = "\n"))
  #scale_x_continuous( limits = c(0,7) )+
  #scale_y_continuous(limits = c(-2,10))+
  #geom_vline(xintercept=1, linetype="dashed", color = "grey")+
  #geom_text_repel(data = Integration_table_uniquePatients, aes(label = Patient_ID))+
  #labs(x ="Ratio of inhibitory marker expression of CD8+ vs. CD4+ T cells", y = "Number of immunogenic MS neoantigens per tumor")

ggscatter(Integration_table_uniquePatients_ImmuTh, x = "freq_CD3", y = "Reactive_neoantigens_perTumor",
          color = "black", shape = 21, size = 3, # Points color, shape and size
          add = "reg.line",  # Add regressin line
          add.params = list(color = "blue", fill = "lightgray"), # Customize reg. line
          conf.int = TRUE, # Add confidence interval
          cor.coef = TRUE, # Add correlation coefficient. see ?stat_cor
          cor.coeff.args = list(method = "spearman", label.x = 0, label.sep = "\n")) +
  scale_x_continuous(breaks = c(0,1,2))
  #labs(x ="Master_Neutrophils", y = "freq_CD3")
  #geom_text_repel(data = Integration_table_uniquePatients, aes(label = Sample_ID))

# _____________ correlation matirx linear correlations __________________#####
corDT # from calculations
corDT_p #from calculations

corDT_rownames <- corDT$rn
corDT_matrix <- corDT
corDT_matrix$rn <- NULL
corDT_matrix[is.na(corDT_matrix)] <- 0
corDT_matrix <- as.matrix(corDT_matrix, rownames = corDT_rownames)

# filter matrix as wanted
#for revision
corDT_cols <-
c("freq_CD3","freq_CD8","CD8_Inhib_no_freq_parent","CD8_Inhib_yes_freq_parent","CD8_Teff_freq_total","CD8_Teff_Inhib_no_freq_parent","CD
8_Teff_Inhib_yes_freq_parent","CD8_Tem_freq_total","CD8_Tem_Inhib_no_freq_parent",
"CD8_Tem_Inhib_yes_freq_parent","freq_CD4","wt_peptidome_1FDR")
corDT_rows <- c("peptides_MS_yes_maybe","peptides_MS_no", "peptides_MS_reactive")
corDT_matrix <- corDT_matrix[corDT_rows,]
corDT_matrix <- corDT_matrix[,corDT_cols]

#rename columns
colnames(corDT_matrix) <- c("CD3 freq","CD8 freq","CD8 InhibMarker yes","CD8 ratio InhibMarker yes/no","CD8 Tn InhibMarker no","CD8 Tn
ActivMarker no","CD8 Teff freq","CD8 Teff ratio InhibMarker yes/no","CD8 Tem freq","CD8 Tem InhibMarker yes","CD8 Tem ratio InhibMarker
yes/no","CD8 Trm freq","CD4 freq","CD8 ratio marker Inhib/Activ yes","Ratio CD8 InhibMarker yes/CD4 InhibMarker yes","Mutations
total","Mutations somatic","Mutations RNAediting","wt peptidome 1%FDR")
rownames(corDT_matrix) <- c("Neoantigens Predicition 9mers","Neoantigens MS", "Reactive neoantigens MS")

corDT_matrix <- corDT_matrix[7:116,] # all phenotyping and mutLoad and peptidome data
corDT_matrix <- corDT_matrix[,117:124] # only neoantigen data

corDT_matrix <-corDT_matrix[,-111:-119]
corDT_matrix <-corDT_matrix[-111:-119,]
corDT_matrix <- corDT_matrix[-83:-98,]
corDT_matrix <- corDT_matrix[,-83:-98]
corDT_matrix <-corDT_matrix[-17:-48,]
corDT_matrix <-corDT_matrix[,-19:-35]

corrplot(corDT_matrix, method="circle", tl.col="black", tl.cex = 0.5) #order="hclust")

#calculate p values
corDT_p_mat <- corDT_p
corDT_p_mat$rn <- NULL
corDT_p_mat <- as.matrix(corDT_p_mat, rownames = corDT_rownames)
mode(corDT_p_mat) <- "numeric"
```

```
# filter matrix
corDT_p_mat <- corDT_p_mat[corDT_rows,]
corDT_p_mat <- corDT_p_mat[,corDT_cols]
colnames(corDT_p_mat) <- c("CD3 freq","CD8 freq","CD8 InhibMarker yes","CD8 ratio InhibMarker yes/no","CD8 Tn InhibMarker no","CD8 Tn
ActivMarker no","CD8 Teff freq","CD8 Teff ratio InhibMarker yes/no","CD8 Tem freq","CD8 Tem InhibMarker yes","CD8 Tem ratio InhibMarker
yes/no","CD8 Trm freq","CD4 freq","CD8 ratio marker Inhib/Activ yes","Ratio CD8 InhibMarker yes/CD4 InhibMarker yes","Mutations
total","Mutations somatic","Mutations RNAediting","wt peptidome 1%FDR")
rownames(corDT_p_mat) <- c("Neoantigens Predicition 9mers","Neoantigens MS", "Reactive neoantigens MS")

corDT_p_mat <- corDT_p_mat[7:116,]
corDT_p_mat <- corDT_p_mat[,117:124]
corDT_p_mat <- corDT_p_mat[,-111:-119]
corDT_p_mat <- corDT_p_mat[-111:-119,]
corDT_p_mat <- corDT_p_mat[-83:-98,]
corDT_p_mat <- corDT_p_mat[,-83:-98]
corDT_p_mat <- corDT_p_mat[-17:-48,]
corDT_p_mat <- corDT_p_mat[,-19:-35]

col1 <- colorRampPalette(c("#67001F", "#B2182B", "#D6604D", "#F4A582", # normal colour scale red to blue
                           "#FDDBC7", "#FFFFFF", "#D1E5F0", "#92C5DE",
                           "#4393C3", "#2166AC", "#053061"))
col2 <- colorRampPalette(c("#053061", "#2166AC","#4393C3","#92C5DE", "#D1E5F0",
                           "#FFFFFF","#FDDBC7","#F4A582","#D6604D","#B2182B","#67001F")) # reverse colour scale blue to red

corrplot(corDT_matrix,
        #method = 'number',
        tl.col="black",
        col = col2(100),
        tl.cex = 0.75, #0.75
        diag = TRUE,
        cl.cex = 0.75,cl.ratio = 0.1, tl.srt = 45, # (cl.cex = size of legend text, cl.ratio = distance of test from legend)
        p.mat = corDT_p_mat, sig.level = 0.05, insig = "blank", outline = FALSE)
# _Vtype = "lower",tl.srt = 45,

# _____________ correlation matirx group correlations ___________________#####
Merged_U_test_response <- fread("Group_correlations/U-test_correlations/Utest_merged_response_long.csv")

#delete blood data
Merged_U_test_response <- Merged_U_test_response[-35:-50,]
Merged_U_test_response <- Merged_U_test_response[-117:-132,]

#delete detailed phenotyping data
Merged_U_test_response <- Merged_U_test_response[-10:-31,]
Merged_U_test_response <- Merged_U_test_response[-35:-36,]
Merged_U_test_response <- Merged_U_test_response[-68:-89,]
Merged_U_test_response <- Merged_U_test_response[-93:-94,]
Merged_U_test_response_sig <- Merged_U_test_response[Merged_U_test_response$pval <= 0.051]

ggplot(Merged_U_test_response, aes(x = Response_type, y = Parameter ))+
  geom_point(aes(size = -pval_log), shape = 21, colour = "black", fill = "cornsilk")

ggplot(Merged_U_test_response, aes(x=Response_type, y=Parameter,size= FDR)) +
  geom_point(shape = 21, colour = "black", fill = "grey", alpha=0.5)+     # plot as points
  geom_point(data = Merged_U_test_response_sig, aes(x=Response_type, y= Parameter, color = pval))+
  geom_text(data = Merged_U_test_response_sig, aes(label=round(pval, 3)), hjust=-0.5, size=3) +   # display the value next to the "balloons" only for
sig. values
  scale_radius(range = c(.05, 10), breaks = c(0,0.25,0.5,0.75,0.95,0.99)) +
  scale_color_continuous(name = "significant \np-values")+
  scale_x_discrete(labels = c("Response to IT", "Response to IT post resection"))+
  #scale_fill_continuous(low = "plum1", high = "purple4")+
  #scale_size(range = c(.1, 10), name="FDR")+
  theme_bw() +
  theme(axis.line = element_blank(),           # disable axis lines
        axis.title = element_blank(),          # disable axis titles
        panel.border = element_blank(),        # disable panel border
        panel.grid.major.x = element_blank(),   # disable lines in grid on X-axis
        panel.grid.minor.x = element_blank())   # disable lines in grid on X-axis

# _____________ ROC curves Response _____________________________________#####
ROC_df <- Integration_table_uniquePatients_ImmuTh
ROCit_obj <-
rocit(score=Integration_table_uniquePatients_ImmuTh$Reactive_neoantigens_perTumor,class=Integration_table_uniquePatients_ImmuTh$Respon
se)

plot(ROCit_obj, YIndex = F, values = T, col = c(2,4), legend = F)
```

```
title(main = "wt peptidom 5% FDR",sub = paste("AUC = ", round(ciAUC(ROCit_obj)$AUC, 2), " [", round(ciAUC(ROCit_obj)$lower, 2)," - ",
round(ciAUC(ROCit_obj)$upper, 2), "]", sep=""))

summary(ROCit_obj)
ciAUC(ROCit_obj) #$lower $upper $AUC

# for loop
# for response
dat <- as.data.frame(ROC_df[,7:171])
dat$HLAI_per_tumor <- NULL
dat$CD3_percent_IHC <- NULL
dat$TCR_diversity <- NULL
Res <- matrix(nrow=162, ncol=3)
row.names(Res) <- names(dat)[1:162]
# for response post
dat <- as.data.frame(ROC_df[,7:171])
dat$HLAI_per_tumor <- NULL
dat$CD3_percent_IHC <- NULL
dat$TCR_diversity <- NULL
Res <- matrix(nrow=1, ncol=3)
row.names(Res) <- names(dat)[1:129]

# select path to save plots
pdf("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/25 Integration/Group_correlations/ROC_curves/Resonse_ROC_Curves_V10_new.pdf")

for(i in 1:162){
  ROCit_obj <- rocit(score=dat[,i],class=dat$Response)
  plot(ROCit_obj, YIndex = F, values = T, col = c(2,4), legend = F)
  title(main=names(dat)[i])
  title(sub = paste("AUC = ", round(ciAUC(ROCit_obj)$AUC, 2), " [", round(ciAUC(ROCit_obj)$lower, 2)," - ", round(ciAUC(ROCit_obj)$upper, 2), "]",
sep=""))
  Res[i, ] <- c(round(ciAUC(ROCit_obj)$AUC, 2), round(ciAUC(ROCit_obj)$lower, 2), round(ciAUC(ROCit_obj)$upper, 2))
}
dev.off()

Res <- as.data.frame(Res, row.names = row.names(Res))
names(Res)[1] <- "AUC"
names(Res)[2] <- "conf.int lower"
names(Res)[3] <- "conf.inf upper"
nobs <- nrow(na.omit(dat[, "Tumor_mass", with=F]))

# _____________ Survival Curves analysis _________________________________######
Survival_MD <- fread("Survival_correlations/Table_Integration_V15_new_SurvivalCurves_MD.csv")
Survival_MD[Survival_MD == "x"] <- NA
Survival_MD <- Survival_MD[1:32]

Survival_ID <- fread("Survival_correlations/Table_Integration_V15_new_SurvivalCurves_ID.csv")
Survival_ID[Survival_ID == "x"] <- NA
Survival_ID <- Survival_ID[1:32]

Survival_MASTER <- fread("Survival_correlations/Table_Integration_V15_new_SurvivalCurves_MASTER.csv")
Survival_MASTER[Survival_MASTER == "x"] <- NA
Survival_MASTER <- Survival_MASTER[1:32]

dat <- data.frame(Survival_MASTER)
#for phenotyping included
dat <- as.data.table(dat)
dat <- dat[!Patient_ID %in% c("ImmuNEO-09", "ImmuNEO-14", "ImmuNEO-30", "ImmuNEO-34")]
dat <- as.data.frame(dat)
dat <- data.frame(Survival_MD)
dat <- data.frame(Survival_ID)
dat$wt_peptides_shared <- NULL
dat$Sort<- NULL
dat$peptides_MS_perPatient_yesno <- NULL
dat$Reactive_neoantigens_perPatient_yesno <- NULL
head(dat)

# ________________ standard parameters by median ________________________######
plot(survfit(Surv(time, status) ~ as.numeric(as.numeric(as.character(NKT_quant))>median(as.numeric(as.character(NKT_quant)), na.rm = TRUE)),
        data=dat), lty=1:2,  col=c("red","blue"))
COX <- summary(coxph(Surv(dat$time, dat$status) ~
as.numeric(as.numeric(as.character(dat$NKT_quant))>median(as.numeric(as.character(dat$NKT_quant)), na.rm = TRUE))), data=dat)
title(main = "Frequency of CD8+ \nSurvival since MASTER",sub = paste("HR = ", round(COX$conf.int[1], 2), " [", round(COX$conf.int[3], 2)," - ",
round(COX$conf.int[4], 2), "]"," P=", round(COX$waldtest[3],4), sep=""), ylab = "Survival probability", xlab = "Time")
legend(1, .2, c("< median", "> median"),lty = c(1:2), col=c("red","blue"))
```

```
survdiff(Surv(time, status) ~ as.numeric(as.numeric(as.character(freq_CD8))>median(as.numeric(as.character(freq_CD8)), na.rm = TRUE)),
       data=dat)
summary(coxph(Surv(dat$time, dat$status) ~
as.numeric(as.numeric(as.character(dat$CD8_Tn_freq))>median(as.numeric(as.character(dat$CD8_Tn_freq)), na.rm = TRUE))), data=dat)

#Mutations_somatic
#Mutations_RNAediting
#peptides_prediction_200nM_SBWB
#CD8_Tem_freq
#quant_wt_peptides_1FDR
#peptides_MS_perPatient
#freq_CD8
#Ratio_Inhib_CD8_Teff / Ratio_Activ_CD8_Teff
# CD8_Tem_Inhib_no_freq

# for loop
Res <- matrix(nrow=164, ncol=4)
row.names(Res) <- names(dat)[6:169]

# select path to save plots
#pdf("Z:/AG-Krackhardt/ImmuNEO project/25 Integration/Survival_Curves_MD.pdf")
pdf("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/25 Integration/Survival_correlations/Survival_Curves_MD_V15_new_freq.pdf")
pdf("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/25 Integration/Survival_correlations/Survival_Curves_ID_V15_new_freq.pdf")
pdf("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/25 Integration/Survival_correlations/Survival_Curves_MASTER_V15_new_freq.pdf")

for(i in 6:169){
  plot(survfit(Surv(dat$time, dat$status) ~ as.numeric(as.numeric(as.character(dat[,i]))>median(as.numeric(as.character(dat[,i])), na.rm = TRUE))),
lty=1:2, col=c("red","blue"))
  title(main=names(dat)[i])
  COX <- summary(coxph(Surv(dat$time, dat$status) ~ as.numeric(as.numeric(as.character(dat[,i]))>median(as.numeric(as.character(dat[,i])), na.rm
= TRUE))), data=dat)
  title(sub = paste("HR = ", round(COX$conf.int[1], 2), " [", round(COX$conf.int[3], 2)," - ", round(COX$conf.int[4], 2), "]", sep=""))
  legend(10, .2, c("< median", "> median"),lty = c(1:2), col=c("red","blue"))
  Res[i-5, ] <- c(round(COX$conf.int[1], 2), round(COX$conf.int[3], 2), round(COX$conf.int[4], 2), round(COX$waldtest[3],4))
}
dev.off()

Res <- as.data.frame(Res, row.names = row.names(Res))
names(Res)[1] <- "HR"
names(Res)[2] <- "conf.int lower"
names(Res)[3] <- "conf.inf upper"
names(Res)[4] <- "p.value.Wald"

# ________________ ratio parameters by >1< _______________________________######
# for phenotyping ratios (><1) or marker parent frquencies (><50)
#single plots
plot(survfit(Surv(time, status) ~ as.numeric(as.numeric(as.character(CD8_Teff_Inhib_yes_freq_parent))>50),
         data=dat), lty=1:2,  col=c("red","blue"))
COX <- summary(coxph(Surv(dat$time, dat$status) ~ as.numeric(as.numeric(as.character(dat$CD8_Teff_Inhib_yes_freq_parent))>50)), data=dat)
title(main = "CD4_Activ_no_freq_parent \nSurvival since ID",sub = paste("HR = ", round(COX$conf.int[1], 2), " [", round(COX$conf.int[3], 2)," - ",
round(COX$conf.int[4], 2), "]","p=", round(COX$waldtest[3],4), sep=""), ylab = "Survival probability", xlab = "Time")
legend(1, .2, c("< 1", ">1"),lty = c(1:2), col=c("red","blue"))

survdiff(Surv(time, status) ~  as.numeric(as.numeric(as.character(dat$CD4_Activ_no_freq_parent))>50), data=dat)
summary(coxph(Surv(dat$time, dat$status) ~ as.numeric(as.numeric(as.character(dat$CD4_Activ_no_freq_parent))>50)), data=dat)

#for-loop
dat <- data.frame(Survival_MASTER)
dat <- data.frame(Survival_MD)
dat <- data.frame(Survival_ID)

cols <- grep("Ratio_|time|status", names(dat), value=T)
dat <- as.data.table(dat)
dat <- dat[, cols, with=FALSE]
dat <- as.data.frame(dat)

Res <- matrix(nrow=18, ncol=4)
row.names(Res) <- names(dat)[3:20]

# select path to save plots
pdf("/Volumes/3m0/AG-Krackhardt/ImmuNEO project/25 Integration/Survival_correlations/Survival_Curves_Ratios_MD_V15_new_freq.pdf")

for(i in 3:20){
  plot(survfit(Surv(dat$time, dat$status) ~ as.numeric(as.numeric(as.character(dat[,i]))>1), data = dat), lty=1:2, col=c("red","blue"))
  title(main=names(dat)[i])
  COX <- summary(coxph(Surv(dat$time, dat$status) ~ as.numeric(as.numeric(as.character(dat[,i]))>1)), data=dat)
  title(sub = paste("HR = ", round(COX$conf.int[1], 2), " [", round(COX$conf.int[3], 2)," - ", round(COX$conf.int[4], 2), "]", sep=""))
  legend(10, .2, c("< 1", "> 1"),lty = c(1:2), col=c("red","blue"))
  Res[i-2, ] <- c(round(COX$conf.int[1], 2), round(COX$conf.int[3], 2), round(COX$conf.int[4], 2), round(COX$waldtest[3],4))
}
dev.off()

Res <- as.data.frame(Res, row.names = row.names(Res))
names(Res)[1] <- "HR"
names(Res)[2] <- "conf.int lower"
names(Res)[3] <- "conf.inf upper"
names(Res)[4] <- "p.value.Wald"

# ________________ binary parameters by yes/no __________________________######
# single plots for yes/no  features
dat <- data.frame(Survival_MASTER)
dat <- data.frame(Survival_MD)
dat <- data.frame(Survival_ID)

plot(survfit(Surv(time, status) ~ as.numeric(as.numeric(as.character(Mut_X_87703729_T_C_RNA)), na.rm = TRUE),
            data=dat), lty=1:2,  col=c("red","blue"))
COX <- summary(coxph(Surv(dat$time, dat$status) ~ as.numeric(as.numeric(as.character(dat$Mut_X_87703729_T_C_RNA)), na.rm = TRUE)),
data=dat)
title(main = "Mut_X_87703729_T_C_RNA \nSurvival since ID",sub = paste("HR = ", round(COX$conf.int[1], 2), " [", round(COX$conf.int[3], 2)," - ",
round(COX$conf.int[4], 2), "]", sep=""), ylab = "Survival probability", xlab = "Time")
legend(5, .2, c("no", "yes"),lty = c(1:2), col=c("red","blue"))

survdiff(Surv(time, status) ~ as.numeric(as.numeric(as.character(Mut_10_47970367_G_A_DNA)), na.rm = TRUE), data=dat)
summary(coxph(Surv(time, status) ~ as.numeric(as.numeric(as.character(Shared_somaticMut_13))>median(as.numeric(as.character(freq_CD8)),
na.rm = TRUE)), data=dat))

# ____________ correlation matrix survival correlation ________________######
# __________________ all __________________________________________________#####
Survival_merged <- fread("Survival_correlations/20220323_Survivals_all_combined_V15_ratios_freq.csv")
row_order <- as.character(unique(Survival_merged$Parameter))

Survival_merged$Parameter <-gsub("Mutations_RNAediting", "Mutations_RNAalterations", Survival_merged$Parameter )
Survival_merged$Parameter <-gsub("TMB_probe_rescued", "Mutations_somatic_perMegabase", Survival_merged$Parameter )
Survival_merged <- Survival_merged[,1:7]
Survival_merged <- Survival_merged[!Survival_merged$Parameter_group %in% c("Genomic data", "Peptidomic data")]

Survival_merged_sig <- Survival_merged[p.value.Wald <= 0.05]
Survival_merged_trend <- Survival_merged[p.value.Wald <= 0.065]

ggplot(Survival_merged, aes(y = Parameter, x = HR,xmin = conf.int_lower, xmax=conf.inf_upper ))+
  geom_point(color = 'grey42')+
  geom_errorbarh(height=.4, color = 'grey42')+
  scale_x_log10(name = "Hazard Ratio", labels=comma)+
  scale_y_discrete(limits = rev)+
  facet_grid(factor(Parameter_group, levels=c('Genomic data','Peptidomic data','CD8 T cells','CD4 T cells', 'Other immune cells'))~Survival_time,
scale= "free", space = "free", switch = "y", labeller = label_wrap_gen(width=10))+ #space="free", #label_wrap_gen(width=10) defines width of label
box
  geom_vline(xintercept=1, color="black", linetype="dashed", alpha=.5)+
  geom_point(data = Survival_merged_trend,aes(x=HR, y=Parameter, colour = Highlight), size=2, color='steelblue1')+
  geom_errorbarh(data = Survival_merged_trend,aes(x=HR, y=Parameter, colour = Highlight), height=.4, color='steelblue1')+
  geom_point(data = Survival_merged_sig,aes(x=HR, y=Parameter, colour = p.value.Wald), size=4, color='dodgerblue3')+
  geom_errorbarh(data = Survival_merged_sig,aes(x=HR, y=Parameter, colour = p.value.Wald), height=.7, color='dodgerblue3', size = 1)+ #add
another layer of coloured points ontop of general plot and adapt size of these points
  theme_bw()+
  theme(legend.position = "none",
        panel.grid.minor = element_blank(),
        axis.text.x = element_text(size= 13),
        axis.text.y = element_text(size= 13),
        plot.title = element_blank(),
        axis.title.y = element_blank(),
        axis.title.x = element_text(size=20),
        strip.text.x = element_text(size = 15, face = "bold"),
        strip.text.y = element_text(size = 13, face = "bold"),
        strip.text.y.left = element_text(angle = 90))

# __________________ phenotyping only ____________________________________#######
Survival_merged <- fread("Survival_correlations/20220704_Survivals_phenotyping_lesspatients_V15_ratios_freq.csv")
row_order <- as.character(unique(Survival_merged$Parameter))

Survival_merged <- Survival_merged[,1:7]
Survival_merged_sig <- Survival_merged[p.value.Wald <= 0.05]
Survival_merged_trend <- Survival_merged[p.value.Wald <= 0.065]
```

```
ggplot(Survival_merged, aes(y = Parameter, x = HR,xmin = conf.int_lower, xmax=conf.inf_upper ))+
  geom_point(color = 'grey42')+
  geom_errorbarh(height=.4, color = 'grey42')+
  scale_x_log10(name = "Hazard Ratio", labels=comma)+
  scale_y_discrete(limits = rev)+
  facet_grid(factor(Parameter_group, levels=c('CD8 T cells','CD4 T cells', 'Others'))~Survival_time, scale= "free", space = "free", switch = "y", labeller =
label_wrap_gen(width=10))+ #space="free", #label_wrap_gen(width=10) defines width of label box
  geom_vline(xintercept=1, color="black", linetype="dashed", alpha=.5)+
  #geom_point(data = Survival_merged_trend,aes(x=HR, y=Parameter, colour = Highlight), size=2, color='steelblue1')+
  #geom_errorbarh(data = Survival_merged_trend,aes(x=HR, y=Parameter, colour = Highlight), height=.4, color='steelblue1')+
  geom_point(data = Survival_merged_sig,aes(x=HR, y=Parameter, colour = p.value.Wald), size=4, color='dodgerblue3')+
  geom_errorbarh(data = Survival_merged_sig,aes(x=HR, y=Parameter, colour = p.value.Wald), height=.7, color='dodgerblue3', size = 1)+ #add
another layer of coloured points ontop of general plot and adapt size of these points
  theme_bw()+
  theme(legend.position = "none",
        panel.grid.minor = element_blank(),
        axis.text.x = element_text(size= 13),
        axis.text.y = element_text(size= 13),
        plot.title = element_blank(),
        axis.title.y = element_blank(),
        axis.title.x = element_text(size=20),
        strip.text.x = element_text(size = 15, face = "bold"),
        strip.text.y = element_text(size = 13, face = "bold"),
        strip.text.y.left = element_text(angle = 90))

ggsave("Survival_correlations/Plots/Forest_plots/20220310_Forestplot_Survival_Phenotyping_new_freq.pdf", plot = last_plot(), device = "pdf",
width = 15, height = 9)

# ___________________ only MASTER survival and facet by parameter_________######
#Survival_merged_MASTER <- Survival_merged[Survival_merged$Survival_time == "MASTER",]
Survival_merged_MASTER <- fread("Survival_correlations/20211025_Survivals_MASTER_combined.csv")
Survival_merged_MASTER_sig <- Survival_merged_MASTER[Highlight == 1] # select for criteria
Survival_merged_MASTER_trend <- Survival_merged_MASTER[Highlight == 2] # select for criteria

ggplot(Survival_merged_MASTER, aes(y = Parameter, x = HR,xmin = conf.int_lower, xmax=conf.inf_upper ))+
  geom_point(color = 'grey42')+
  geom_errorbarh(height=.4, color = 'grey42')+
  scale_x_log10(name = "Hazard Ratio")+
  scale_y_discrete(limits = rev)+
  facet_grid(Parameter_group~. , scale = "free", space = "free",  switch = "y")+ #space="free"
  geom_vline(xintercept=1, color="black", linetype="dashed", alpha=.5)+
  geom_point(data = Survival_merged_MASTER_sig,aes(x=HR, y=Parameter, colour = Highlight), size=4, color='dodgerblue3')+
  geom_errorbarh(data = Survival_merged_MASTER_sig,aes(x=HR, y=Parameter, colour = Highlight), height=.7, color='dodgerblue3', size = 1)+ #add
another layer of coloured points ontop of general plot and adapt size of these points
  geom_point(data = Survival_merged_MASTER_trend,aes(x=HR, y=Parameter, colour = Highlight), size=2, color='steelblue1')+
  geom_errorbarh(data = Survival_merged_MASTER_trend,aes(x=HR, y=Parameter, colour = Highlight), height=.4, color='steelblue1')+
  theme_bw()+
  theme(legend.position = "none",
        panel.grid.minor = element_blank(),
        axis.text.x = element_text(size= 15),
        axis.text.y = element_text(size= 13),
        plot.title = element_blank(),
        axis.title.y = element_blank(),
        axis.title.x = element_text(size=20),
        strip.text.x = element_text(size = 15, face = "bold"),
        strip.text.y = element_text(size = 13, face = "bold"),
        strip.text.y.left = element_text(angle = 90))

forestplot(labeltext =Survival_merged$Parameter, mean =Survival_merged$HR, lower = Survival_merged$conf.int_lower, upper =
Survival_merged$conf.inf_upper,
           #xlog= TRUE,
           clip = c(-.1, 4.5),
           boxsize = 0.2,
           vertices = TRUE)
```