Technische Universität München
TUM School of Medicine and Health

Identification of neoantigen-specific T-cell responses in diverse rare tumor entities

Philipp Martin Seifert

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

Doktors der Medizin (Dr. med.)

genehmigten Dissertation.

Vorsitz: apl. Prof. Dr. Bernhard Haslinger

Prüfende der Dissertation:

1. Prof. Dr. Angela Krackhardt
2. Priv.-Doz. Dr. Simon Heidegger

Die Dissertation wurde am 06.09.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Medicine and Health am 04.01.2024 angenommen.

Fakultät für Medizin
Technische Universität München, Klinikum rechts der Isar

Doctoral Thesis in Medicine

Identification of neoantigen-specific T-cell responses in diverse rare tumor entities

Erkennung von neoantigenspezifischen T-Zell-Immunantworten in verschiedenen seltenen Tumorentitäten

Author:	Philipp Seifert
Supervisor:	Prof. Dr. Angela Krackhardt
Advisors:	Dr. Eva Bräunlein
Submission Date:	August 10, 2023

Declaration of incorporated publications

Parts of this thesis have already been submitted for publishing:

Celina Tretter*, Niklas de Andrade Krätzig*, Matteo Pecoraro, Sebastian Lange, Thomas Engleitner, Philipp Seifert, Clara von Frankenberg, Johannes Untch, Florian Dreyer, Eva Bräunlein, Mathias Wilhelm, Daniel Zolg, 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 Küster, Matthias Mann, Wilko Weichert, Stefan Fröhling, Roland Rad, Michael Hiltensperger, Angela M Krackhardt. Proteogenomic analysis reveals RNA as an important source for tumor-agnostic neoantigen identification correlating with T-cell infiltration. *Accepted for publication in Nat. Commun. 2023* (*equal contribution)

Abstract

Cancer immunotherapy by checkpoint modulation has shown high efficacy in a large variety of neoplasms and is currently transforming the treatment of malignant diseases. In this context, understanding the nature of immunological tumor recognition or ignorance is of utmost importance to further improve current immunotherapeutic approaches.

As can be seen, i.e., in the process of immunosurveillance, mutated peptide ligands (neoantigens) that can be predicted in silico from cancer genome sequencing data or that can be identified by mass spectrometry approaches may serve as highly tumor-specific target antigens (TSA). Some selected clonal neoantigens may be of particular relevance for prolonged responses. In contrast, otherwise, clonal evolution and immunoediting of neoantigens may be closely related and associated with cross-resistance to targeted drugs and immunotherapy. However, these pipelines are still subject to crucial limitations regarding target specificity and therapeutic feasibility.

By detailed analysis of the mutanome, resulting peptide specifications, and associated HLA binding predictions for a comprehensive proteogenomic data set for 32 patients across 25 tumor types, it could, among other findings, be shown that a significant part of non-wild type HLA-binding peptides results from variants identified on aberrantly expressed RNA transcripts. This not only underlines the importance of RNA-centered variant detection but may, consequently, notably amplify the available neoepitope repertoire, eventually resulting in the adaption of pre-existing identification pipelines. Furthermore, an immunogenicity assessment assay was used to measure neoantigen-dependent interferon-based T-cell responses in patient blood samples to screen for possible target candidates.

Successful characterization and classification of potential neoantigens will greatly enhance our understanding of tumor immune system interactions, leading to the identification of novel biomarkers and effective combinatorial strategies in immuno-oncology.

Zusammenfassung

Die Behandlung von Krebserkrankungen durch Immuntherapien, insbesondere mittels Checkpoint Modulation, hat in verschiedenen Studien hohe Wirksamkeit gezeigt und verändert gerade grundlegend die bisher etablierten Therapiekonzepte. Ein umfassendes Verständnis der Tumorerkennung durch das menschliche Immunsystem ist daher von höchster Bedeutung, um aktuelle Ansätze in der Immuntherapie weiter zu verbessern.

Die genaue Betrachtung von Prozessen wie der "Immunosurveillance" zeigt, dass mutierte Peptidliganden (Neoantigene), die aus Krebsgenomsequenzierungsdaten in silico vorhergesagt oder durch Massenspektrometrie identifiziert werden können, als hoch tumorspezifische Zielantigene dienen können. Einige klonale Neoantigene sind dabei von besonderer Bedeutung, da sie möglicherweise langanhaltende tumorspezifische Immunantworten hervorrufen, welche therapeutischen Nutzen besitzen. Jedoch können klonale Evolution und Immunoediting spezielle Resistenzen gegenüber zielgerichteten Medikamenten im Rahmen der Immuntherapie hervorrufen. Aktuelle Verfahren zur Entdeckung solcher Neoantigene haben daher noch einige Limitationen, unter anderem eine unzureichende Spezifität sowie therapeutische Relevanz im klinischen Kontext.

Mithilfe des Mutanoms, der Eigenschaften der daraus resultierenden Peptide sowie der Simulation ihrer Bindungseigenschaften konnten wichtige Merkmale der durch MHC Klasse I präsentierten Peptide sowie deren genetische Ursachen untersucht werden. Die Auswertung von Datensätzen von 25 verschiedenen Tumorarten, die von 32 Patienten erhoben wurden, zeigte die herausragende Bedeutung von nicht kanonisch entstandenen, aberrant exprimierten RNA-Transkripten. Diese Ergebnisse unterstreichen nicht nur die Wichtigkeit der Sequenzierung von RNA-Daten, sondern auch, dass das Repertoire an Neoantigenen möglicherweise durch umfassendere nicht exonische Analysen noch deutlich erweitert werden kann. Daraus ergeben sich Anpassungen der Pipelines zur

Identifizierung von Neoantigenen.

Darüber hinaus wurden Interferon-basierte in vitro Stimulationsversuche durchgeführt, um immunogene spezifische T-Zell-Klone gegen einzelne Neoantigene zu detektieren und zu charakterisieren. Auf diese Weise sollten potenzielle Zielstrukturen auf ihre Immunogenität getestet werden.

Eine erfolgreiche Charakterisierung und Klassifizierung von Neoantigenen und der dadurch ausgelösten Immunantworten ist notwendig, um unser Verständnis der Interaktion von Tumor und Immunsystem zu verbessern. Hierdurch kann die Entdeckung neuer Biomarker, sowie die Entwicklung effektiver kombinatorischer Behandlungsstrategien entschieden vorangebracht werden.

Contents

1 Introduction

1.1 Immunotherapy as an integral part of cancer treatment strategies

When William B. Coley started injecting mixtures of live and inactivated Streptococcus bacteria to patients with bone sarcomas in 1891 to induce sepsis and anti-tumor responses, he was maybe not aware of performing the first ever documented active cancer immunotherapy intervention (Esfahani et al., 2020). More than 120 years later, the anti-CTLA4 antibody Ipilimumab was approved, marking a new era for cancer immunotherapies. For the first time, a therapy could significantly increase the survival rate of patients with metastatic melanoma, a feat that had been impossible with conventional treatments (Mellman et al., 2011). Concomitantly many different immunotherapeutic regimes such as cancer vaccines, oncolytic viruses, adoptive transfer of activated T cells, and antibodies blocking immune-checkpoint pathways have shown to be highly promising treatment modalities for a variety of cancer entities (Farkona et al., 2016).

All of these approaches aim to modulate the immune system such that malignant cells are not only considered natural targets but, as a result, can be detected and attacked by cells of the immune system. This idea traces back to the concept of tumor immunogenicity, which states that proteins expressed in cancer cells may function as tumor antigens and trigger an immune response (Smyth et al., 2006). However, the interaction between the immune system and cancer cells within the tumor microenvironment is complex. It may not only lead to tumor destruction (known as "immunosurveillance") but can also promote further tumor growth (Bui & Schreiber, 2007). These two phenomena, generally described as "immunoediting," are currently subject of intense research (O'Donnell et al., 2019).

In many ways unscrambling the interplay of cancer and the immune system remains a highly complex challenge, yet many successful implementations of different immunotherapies can be mentioned (Hegde & Chen, 2020). The following will briefly outline the most widely applied and promising immunotherapeutic modalities.

1.1.1 Therapies with monoclonal antibodies

One of the most prominent tools when using the immune system to target cancer cells are monoclonal antibodys (mAbs). These proteins consist of a Fab and an Fc region, which allows them to bind possible targets as well as components of the immune system with high affinity and specificity. Thus, mAbs can mediate immune responses helping to attack and destroy cancer cells (Kimiz-Gebologlu et al., 2018). Depending on the specific antibody subtype, a variety of different mechanisms with specialized functions such as antibody dependent cellular cytotoxicity (ADCC) and complement-dependent cytotoxicity (CDC) can be distinguished (Scott et al., 2012; Zahavi & Weiner, 2020).

Monoclonal antibodies, as used in nowadays therapies, are clonal versions of a specified antibody isotype directed against a specific target antigen. Different examples illustrate the various mechanisms that can be utilized to reduce tumor growth.
Cetuximab is an antibody against epidermal growth factor receptor (EGFR), which, when up-regulated, is associated with tumor cell proliferation. By blocking the ligand binding and receptor dimerization, Cetuximab induces apoptosis of the target cell. (Li et al., 2005). It can be used for cancers with overexpression of EGFR, such as colorectal cancer. In contrast, Trastuzumab, an anti-human epidermal growth factor receptor 2 (HER2)-antibody is used to treat HER2-amplified breast cancers by inhibiting receptor hetero-dimerization and consequential signal perturbation (Chen et al., 2003).
Rituximab, a widely applied mAbs already approved in 1997, utilizes more indirect mechanisms. It targets CD20-positive B cells through CDC, ADCC, and antibody-dependent cellular phagocytosis (ADCP) and is used to treat B-cell malignancies, including diffuse large B-cell lymphoma, follicular lymphoma and chronic lymphocytic leukemia (Salles et al., 2017; Pierpont et al., 2018).

Beyond the previously mentioned areas in which mAbs can be utilized for cancer therapy, the tumor microenvironment provides additional valuable targets for mAbs-based

immunotherapy. Different mAb were designed to inhibit the up-regulation of tumor angiogenesis by either directly targeting and deactivating **VEGF!**s (**VEGF!**s) (Bevacizumab) or by blocking the **VEGF!** receptor (Ramucirumab) (Sullivan & Brekken, 2010).

Using mAbs in antibody-drug conjugates (ADCs) represents another therapeutic approach in cancer immunotherapy. Here the antibody is utilized to deliver either radionuclides, biological toxins, or ultratoxic payload as agents for direct internalization in target cancer cells (Chau et al., 2019). Since showing impressive activity against treatment-refractory cancers, different ADCs are currently approved for cancer treatment (Drago et al., 2021). Brentuximab vedotin, for example, has proven to be beneficial in some cases of stage III or IV of Hodgkin's Lymphoma (Ansell et al., 2022) and Trastuzumab emtansine, Inotuzumab ozogamicin and Enfortumab vedotin find successful application Her2-positive breast cancer, acute lymphoblastic leukemia, and bladder cancer (von Minckwitz et al., 2019; Bhojwani et al., 2019; Powles et al., 2021).

1.1.2 Therapies with antibodies modulating the immune response

Besides the previously described use of mAbs to target cancer cell epitopes, another anti-tumor strategy involves targeting immune cells to boost immune responses (Zahavi & Weiner, 2020). A relatively new but nowadays widespread approach is the inhibition of immune checkpoints, which are control pathways that regulate immune responses to prevent tissue damage and maintain self-tolerance. To prevent cancer cells from exploiting these checkpoints to down-regulate the immune response, mAbs, so-called immune checkpoint inhibitors, can be used to target inhibitory checkpoints, thereby enhancing T-cell activation (Pardoll, 2012). The blockade of cytotoxic T-lymphocyte-associated antigen 4 (CTLA4) with Ipilimumab, i.e., which was the first identified checkpoint inhibitor approved for cancer treatment in 2011, showed immensely successful results in the treatment of metastatic melanoma patients (Hodi et al., 2010). In 2015, Nivolumab and Pembrolizumab were the first checkpoint inhibitors used to treat advanced non-small-cell lung cancer (NSCLC) (Borghaei et al., 2015; Garon et al., 2015), followed by others such as Atezolizumab (2016), Avelumab and Durvalumab (2017) which were also applied for the treatment of head and neck cancer, bladder cancer, Merkel cell cancer as well as classic Hodgkin's lymphoma (Rotte et al., 2018; Inman et al., 2017; Socinski et al., 2018; Antonia

et al., 2018; D'Angelo et al., 2018). By blocking programmed cell death 1 receptor (PD-1) or its ligand programmed cell death receptor ligand 1 (PD-L1), both part of the PD-1 signaling pathway, these mAbs inhibit down-regulation of T cell-mediated immune responses and thus can avoid tumor immune evasion (Akinleye & Rasool, 2019). Besides PD-1 and CTLA4, lymphocyte-activated gene 3 (LAG-3), another group of cell surface inhibitory receptors, is being evaluated at various stages of pre-clinical and clinical development (Chocarro et al., 2022; Andrews et al., 2017). Moreover, a variety of other targets and their respective pathways, such as T cell immunoreceptor with Ig and ITIM domain (TIGIT) or **VISTA!** (**VISTA!**), are currently under investigation not only as potential therapeutic targets but also as prognostic markers and biomarkers for immunotherapy (Chauvin & Zarour, 2020; Tagliamento et al., 2021).

1.1.3 Cell-based therapies

Another immunotherapeutic approach uses autologous or allogenic T cells that recognize specific structures on the surface of tumor cells. Earlier attempts during the 1980s where for the first time tumor-infiltrating lymphocytes (TILs) were extracted, expanded, and reinfused (Rosenberg et al., 1994), have demonstrated encouraging efficacy and led to modern TIL-based approaches that are nowadays tested in Phase 2 clinical trials (Schoenfeld et al., 2021; O'Malley et al., 2021). However, the most notable clinical progress with adoptive cell therapy (ACT) began with the development of receptor-engineered lymphocytes (Waldman et al., 2020). Unlike conventional T cells, these therapies have the crucial advantage in that they bypass MHC-restrictions by using a synthetic chimeric antigen receptor (CAR). This receptor can directly detect target molecules on the surface of malignant cells, which allows it to bind cancer cells even after having downregulated MHC expression. This ability of CARs to form non-classical immune synapses facilitates the mediation of anti-tumoral effects via perforin, granzymes, and cytokines (Benmebarek et al., 2019). By avoiding the classical MHC-antigen binding of normal T cells, usually required for any canonical T-cell based immune response (Zareie et al., 2021), approaches with CAR T cells targeting CD19 showed promising efficacy in the treatment of B cell acute lymphoblastic leukaemia (B-ALL) (Maude et al., 2014), multiple subtypes of B-cell lymphoma (Brudno & Kochenderfer, 2018) and in the treatment of patients with relapsed or refractory mantle-cell lymphoma (Wang et al., 2020). Newer trials with

CAR T cells targeting B cell maturation antigen (BCMA), which is expressed on plasma cells, even indicate promising results in patients with multiple myeloma (Mikkilineni & Kochenderfer, 2021; Raje et al., 2019). Moreover, CD30 CAR T cells have been successfully used for the treatment of relapsed and refractory Hodgkin Lymphoma (Ramos et al., 2020). The pool of potential targets for various cancer types is continuously growing as an increasing number of CAR T cell therapies are subject to clinical investigations.

Besides the potential and recent success, severe toxicity, restricted trafficking, limited tumor infiltration, reduced activation of CAR T cells within solid tumors, and finally, tumor heterogeneity with antigen escape are just some of the key challenges that still remain and that need to be addressed to overcome therapeutic limitations (Bonifant et al., 2016; Majzner & Mackall, 2018; Rafiq et al., 2020). In addition, interactions between CAR T cells and the **TME!** (**TME!**) play a critical role since they have been associated with limited local accessibility of tumor cells and reduced functionality of CAR T cells (Rafiq et al., 2020; Sterner & Sterner, 2021).

1.1.4 Cancer vaccines

The principles of vaccination against different diseases caused by bacteria or viruses have been well-established for many decades. Not only are there vaccines against many different pathogens available, but also is their use indispensable in terms of preventive health care (Bonanni, 1999).
Even cancer-prophylactic vaccines targeting certain oncoviruses, i.e. human papillomavirus (HPV) and hepatitis B virus (HBV) that are directly involved in tumor pathogenesis are nowadays widely disposed (Lowy & Schiller, 2006; Arzumanyan et al., 2013). Immunization programs with these vaccines contribute substantially to preventing different malignant diseases such as cervical or liver cancer (Falcaro et al., 2021; McGlynn et al., 2021).

There are three broad categories of tumor antigens: tumor associated antigens (TAAs), cancer testis antigens (CTAs), and **TSA!**s (**TSA!**s). TAAs are proteins that are also encoded in the normal genome and that are hence present in different tissues. Using them as therapy targets requires an abnormally high expression level on tumor cells to circumvent central tolerance (Hogquist et al., 2005). Although CTAs are also encoded in normal

cells, their expression is restricted to certain germ cell tissue and cancer cells (Simpson et al., 2005). In contrast, **TSA!**s are not encoded in the normal host genome but arise, amongst others, as a consequence of tumor-specific mutations (Schumacher & Schreiber, 2015; Gubin et al., 2015). These novel protein sequences that are highly specific for tumor cells cannot be found on healthy tissue and are therefore less susceptible to mechanisms of immunological tolerance, making them promising targets for immunotherapy (Heemskerk et al., 2013).

The concept of a therapeutic vaccine against tumor diseases is similar to a classical vaccine but requires specific target structures on the surface of the tumor cell. In contrast to TAAs, which were shown to function as possible vaccination targets already in the 1970s (Hanna & Peters, 1978) and that can also be expressed on healthy tissues, it has been provided firm evidence that **TSA!**s not only induce more specific but also more robust immune responses (Jiang et al., 2019). Tumor-restricted expression of these targets ensures that there are no adverse effects due to on-target, off-tumor damage to healthy tissue caused by unspecific T cell reactivity (Schumacher et al., 2019).

In 2012, Castle et al. could show that by identifying nonsynonymous somatic point mutations in murine melanoma cells with next-generation sequencing (NGS), they could not only provide possible neoantigen targets for vaccination but also prove immunogenicity for a substantial fraction of neoantigens within a mouse model (Castle et al., 2012). Newer trials suggest that major histocompatibility complex (MHC) class II-restricted neoantigens have a key function in the activation and the successful interplay between $CD8^+$ and $CD4^+$ T cells, beneficial for establishing anti-tumor responses (Alspach et al., 2019).

Moreover, clinical trials have shown that by injecting high-risk Melanoma patients with ribonucleic acid (RNA) encoding for individual mutations, $CD4^+$ and $CD8^+$ T-cell responses could be induced that led to an increased progression-free survival (Sahin & Türeci, 2018). The identification of the required genetic variants was based on NGS and the help of MHC class I and class II binding predictions. Similar results could be observed in different studies, repeatedly pointing out the beneficial outcome of a subsequent immune checkpoint inhibition (ICI) therapy (with anti-PD-1) in case of progression, associated with an expanded repertoire of neoantigen-specific T cells (Ott et al., 2017; Sahin & Türeci, 2018). Current studies are investigating the synergistic effects between vaccination and ICI, the selection criteria of efficient neoepitopes, and their deliv-

ery platform (Saxena et al., 2021).

1.2 Limitations of immunotherapy

1.2.1 Mechanisms of Resistance

Understanding the mechanisms of resistance used by different malignancies to escape the immune system is indispensable for developing novel strategies to overcome the current limitations of immunotherapy.

Recent NGS analyses of a vast number of tumors indicate remarkable differences for intratumor heterogeneity (ITH) for different cancers. In parts, this can be explained by different genetic and epigenetic processes, such as somatic mutations or DNA methylation, that underlie carcinogenesis (Alexandrov et al., 2013a). The resulting clonal diversity is linked to therapeutic failure, drug resistance, and poor clinical outcome (Mazor et al., 2016; McGranahan & Swanton, 2017).

A widely known mechanism of immunoevasion is based on the ability of cancer cells to inactivate or epigenetically regulate the expression of genes encoding antigen-processing components, leading to diminished MHC class I expression (Sade-Feldman et al., 2017). By reducing antigen-processing and presentation (APP), the tumor escapes immunosurveillance. It avoids elimination by $CD8^+$ T cells, resulting in reduced response rates to therapies that target MHC-class I restricted antigens, such as ICI (Dhatchinamoorthy et al., 2021).

To address this problem, MHC class-I independent targeting strategies, such as CAR T cells, have been developed. However, those approaches are faced with various challenges, such as antigen escape or lineage switch, which are also a direct result of ITH and have been observed in $CD19^+$-CAR T cell therapies (Majzner & Mackall, 2018).

1.2.2 Adverse effects of cancer immunotherapy

Despite an increasing number of successful immunotherapeutic treatments and continuous improvements in different related clinical areas, a notably high fraction of patients

treated, i.e., with ICI experience significant immune-related adverse events (irAEs). Typical adverse effects range from mild skin abnormalities to severe impacts on multiple organs of the body, such as the GI tract, the lungs, the liver, the renal system, the endocrine glands, or the cardiovascular and ocular systems (Michot et al., 2016; Brahmer et al., 2018).

A diversity of mechanisms contribute to the genesis of these side effects. These include exacerbation of pre-existing autoimmunity, such as by certain HLA haplotypes or single nucleotide polymorphisms (SNPs), aberrant presentation of generally restricted self-peptides, such as CTAs, or loss of tolerance driven by the **TME!** (Burke et al., 2020; Weinmann & Pisetsky, 2019). Although adverse effects may result from a combination of these mechanisms, there is evidence suggesting a connection between anti-tumor immunity and autoimmunity, with shared transcriptional signatures present in both processes (Schnell et al., 2020). The differences, but also the similarities in the process of immunoregulation could be a key starting point to enhance anti-tumor immunity without increasing autoimmunity.

As the field of immunotherapy keeps evolving, clinicians will have to face the consequences of irAEs, which require a high level of awareness and skilled management to ensure best patient outcomes (Martins et al., 2019). Moreover, advanced biomaterials and drug delivery systems could help to effectively control the therapy application, improve their potency and thereby reduce toxic side effects (Riley et al., 2019). Although good techniques for controlling irAEs have been established in the last years, irAEs remain a key challenge in the clinical setting and will encourage to advance towards the development of more specific immunotherapies.

1.3 The significance of neoantigens in immunotherapy

1.3.1 A variety of sources for possible specific target antigens

Although immunotherapeutic approaches have proven effective in a wide range of human malignancies, the therapeutic potential of T-cell-based techniques has yet to be fully explored (Leko & Rosenberg, 2020). The activation of T cells and, thus, the effectiveness of therapy critically depends on recognizing cancer peptide epitopes and their ability to induce cancer rejection mechanisms. As was shown by Tumeh et al. (2014) for PD-1 ICI exploiting pre-existing, but negatively regulated $CD8^+$ T cell repertoires is of central importance to induce desired cytotoxic T-cell activities.

The mechanisms responsible for the emergence of neoantigens are diverse. They can include somatic mutations (i.e. DNA variants), chromosomal aberrations, or other non-canonical products such as defective ribosomal products (DRiPs), that are presented on MHC class-I molecules (Dersh et al., 2021). These aberrantly expressed proteins are proposed to constitute an important additional source of tumor neoantigens (Laumont et al., 2016).

The exact rules for endogenous antigen presentation, including processes like protein cleavage and gene expression, are still the subject of intensive studies (Villani et al., 2018). Further experimental validation of predicted neoepitopes is needed, as only a tiny subset of peptides will be processed and presented in the context of MHC on the cell surface (Vitiello & Zanetti, 2017).

1.3.2 Immunoinformatics and the rules of antigen presentation

Presentation of antigens on MHC molecules represents a fundamental principle of immunotherapy and is necessary for any target recognition by T cells. However, the underlying selection criteria or rules for MHC presentation remain largely unknown, which is a result of the highly polygenic and polymorphic nature of the MHC, giving rise to a particular set of molecules with individual peptide-binding specificities (Abualrous et al., 2021). This enormous variety of MHC allotypes mapped to the individually unique peptide landscape requires sophisticated immunoinformatics for binding prediction and

subsequent epitope discovery (Nielsen et al., 2020).

Structural features of the MHC-specific binding groove regulate the formation of peptide MHC (pMHC) complexes, a process modified by two peptide editors. They shape the presented peptidome, thereby favoring the binding of high-affinity antigens (Wieczorek et al., 2017).

After years of intensive trials of affinity prediction with simple motif-based models, such as the SYFPEITHI prediction model (Rammensee et al., 1999), that were focused on cataloging and classification of different binding motifs, the development of modern machine learning algorithms in the 1990s cleared the way for a variety of entirely new approaches. The idea of these methods is to minimize the error between an experimentally validated training data set and the corresponding predictions of the model, which can subsequently be used to predict new unknown data (Nielsen et al., 2020). Even though different proposed models such as hidden Markov models (HMMs), quantitative structure-affinity relationship (QSAR)-based affinity models, and artificial neural networks (ANNs) initially suffered from low accuracy due to a lack of data, advancements in the field of high-throughput peptide–MHC-binding assays, as well as different publicly available databases, i.e., the immune epitope database (IEDB), led to increasing improvements in their performance (Nielsen et al., 2003; Peters et al., 2006; Vita et al., 2015).

Despite the emergence of a huge variety of competing models with increasingly precise binding predictions in the last years, predicting the MHC ligandome (the peptide repertoire presented by MHC molecules) remains challenging. At this point, mass spectrometry (MS) may provide additional information on naturally presented peptides resulting from mechanisms like proteasomal cleavage or translocation by the transporter associated with antigen processing (TAP) molecule that are needed for profound comprehension of MHC-mediated antigen presentation (Nielsen et al., 2005; Larsen et al., 2005; Caron et al., 2015).

New MS approaches combining experimental and computational techniques facilitated the discovery of MHC ligands and led to the development of eluted ligand (EL) datasets with annotated MHC restrictions (Bassani-Sternberg et al., 2015). The combination of MS based MHC class I presentation likelihood together with quantitative binding affinity predictions of ANNs has led to the development of improved prediction models, such as

NetMHCpan-4.0 and MHCflurry (Jurtz et al., 2017; O'Donnell et al., 2018).

1.3.3 Towards T cell epitope discovery

Despite the increasing efforts to improve current MHC-based prediction algorithms, the greater goal remains the identification of suitable targets eliciting effective anti-tumor T-cell responses. This raises the fundamental question of the nature of neoantigen immunogenicity and the accuracy of above described binding prediction algorithms in forecasting T-cell reactivity. Although previous experiments generally indicate a low rate of experimental validation of in silico predicted neoepitopes, it was claimed that the majority of immunogenic peptides share a strong predicted HLA binding, with a peak in length of 9 amino acids (AAs) (9-mers) (Trolle et al., 2016; Bjerregaard et al., 2017).
Detailed studies of the MHC class I immunopeptidome revealed that the entire MHC class I–associated peptide (MAP) repertoire covers only a small fraction of expressed exomic sequences resulting in a heterogeneous representation of protein-coding genes within the MAP repertoire. This indicates that a substantially high fraction of genes does not produce MAPs, which could, together with limitations in the T_{CD8^+}R-repertoire, explain low response rates for predicted neoepitopes (Yewdell & Bennink, 1999; Pearson et al., 2016).

To address this issue, which has major implications on the feasibility of personalized identification pipelines, it was suggested to combine sequencing methods with MS-methods to select for MHC class I binding peptides. After the demonstration of this work-flow for two widely used murine models (Yadav et al., 2014), mass spectrometry analysis of human leukocyte antigen (HLA) bound peptides from mono-allelic cell lines as well as from human melanoma cells were shown to greatly contribute to improve possible neoantigen discovery pipelines (Abelin et al., 2017; Bassani-Sternberg et al., 2016).

Recent publications even claim that MS represents the only unbiased methodology to comprehensively examine the repertoire of naturally presented HLA binding peptides, furthermore including post-translational modified peptides (Bassani-Sternberg, 2018). However, this increased specificity and thus reduced false discovery rate (FDR) due to MS-based epitope selection directly results in reduced sensitivity since even modern tandem mass spectrometry (MS/MS)-pipelines only reflect a tiny fraction of the MHC lig-

andome. The finite amount of available material, the chemical properties of the epitope, or the lack of suitable antibodies for some HLA alleles are just some of the limitations responsible for a restricted coverage of the immunopeptidome (Purcell et al., 2019).

Most recently, the discovery of MHC class I presented peptides originating from novel or unannotated open reading frames (nuORFs) or from non-coding yet aberrantly expressed RNA transcripts could re-emphasize the importance of MS-based approaches for the identification of immunogenic neoantigens (Laumont et al., 2018; Ouspenskaia et al., 2022). Studies done on proteasome-spliced peptides (PSPs) also claim that a significant part of the HLA ligandome could be explained by alternative tumor-specific splicing events that are known to be far more abundant than somatic single-nucleotide variants (Mylonas et al., 2018; Hoyos & Abdel-Wahab, 2018; Kahles et al., 2018a). Although the extent to which all these RNA-seq-based peptides ultimately contribute to the immunopeptidome is still unknown, their significance as a potential source of neoantigens for cancer immunotherapy might be huge (Erhard et al., 2018; Shen et al., 2019).

Further detailed analysis of RNA-seq data and experimental immunogenicity assessment of potential pipeline-derived neoantigens may provide deeper insight into the underlying mechanisms responsible for the formation of immunogenic cancer neoantigens.

2 Material and Methods

2.1 Material

2.1.1 Technical Equipment

#	Device	Company
1	Analytical balance SI-64	Denver Instrument Sartorius AG, Göttingen, Germany
2	APOLLO Liquid nitrogen vacuum container	Cryotherm, KirchenSieg, Germany
3	Autoclave Systec V95	Systec GmbH, Linden, Germany
4	BIOSAFE MD sample container	Cryotherm, Kirchen/Sieg, Germany
5	Centrifuge 5417R	Eppendorf AG, Hamburg, Germany
6	Centrifuge 5810R	Eppendorf AG, Hamburg, Germany
7	Centrifuge with vortex 7-0040	neoLab Migge GmbH, Heidelberg, Germany
8	EcoVac Vacuum Pump	schuett-biotec GmbH, Göttingen, Germany
9	Growth chamber WTC	BINDER GmbH, Tuttlingen, Germany
10	HERAfreezeTM BASIC -86°C Freezer	Thermo Fisher scientific, Waltham, USA
11	ImmunoSpot S6 Ultra-V Analyzer	CTL - Europe GmbH, Bonn, Germany

12	Incubator BBD 6220	Heraeus Holding GmbH, Hanau, Germany
13	Incubator CB 150	BINDER GmbH, Tuttlingen, Germany
14	Irradiation chamber Cs137 Type Ob 29/902-1	Buchler GmbH, Braunschweig, Germany
15	Laminar flow HERAsafe KS 15	Heraeus Holding GmbH, Hanau, Germany
16	LS6000 sample container	tec-lab GmbH, Taunusstein, Germany
17	Magnetic stirrer RH basic 2	IKA®-Werke GmbH & CO. KG, Staufen, Germay
18	Microscope Axiovert 40 C	Carl Zeiss AG, Feldbach, Schweiz
19	Multichannel pipets	Eppendorf AG, Hamburg, Germany
20	Multifuge 3 S-R	Heraeus Holding GmbH, Hanau, Germany
21	Multifuge 3s	Heraeus Holding GmbH, Hanau, Germany
22	NALGENE Cryo 1°C Freezing Container	Thermo Fisher Scientific, Waltham, USA
23	Neubauer improved counting chamber	Karl Hecht GmbH & Co KG, Sondheim/Röhn, Deutschland
24	Pipets	Eppendorf AG, Hamburg, Germany
25	Pipette controller	INTEGRA Biosciences GmbH, Biebertal, Germany
26	Precision balance 440	KERN & SOHN GmbH, Balingen, Germany
27	Premium -20°C Freezer	Liebherr-International Deutschland GmbH, Biberach an der Riß, Germany
28	Refrigerator Profi line	Liebherr-International Deutschland GmbH, Biberach an der Riß, Germany
29	Rotina 420R	Andreas Hettich GmbH & Co.KG, Tuttlingen, Germany
30	SunriseTM absorbance reader	Tecan Group Ltd., Maännedorf, Switzerland

31	Vortex Mixer 7-2020	neoLab Migge GmbH, Heidelberg, Germany
32	Vortexer Reax top	Heidolph Instruments GmbH & Co.KG, Schwabach, Germany
33	Vortex-Genie 2	Scientific Industries, Inc., New York, USA
34	Waterbath	Memmert GmbH + Co. KG, Schwabach, Germany
35	Ziegra Ice machine	ZIEGRA Eismaschinen GmbH, Isernhagen, Germany

Table 2.1: Devices

2.1.2 Consumables

#	Consumable	Company
1	Cell culture flask (T25, T75, T175)	Greiner Bio-One GmbH, Frickenhausen, Germany
2	CyroPure tubes	Sarstedt AG & Co., Nümbrecht, Germany
3	EIA/RIA plates	Corning, New York, USA
4	Falcons (15ml, 50 ml)	BD Biosciences, Franklin Lakes, USA
5	Filcon 30 μm filter	Syntec International, Dublin, Ireland
6	Gloves Dermatril P	KCL GmbH, Eichenzell, Germany
7	MAHAS4510 MultiScreen-HA 0.45 μm ELIspot plate	Merck KGaA, Darmstadt, Germany
8	Microtubes (1.2 ml)	Alpha Laboratories, Hampshire, UK
9	neoScrew Micro tubes 1.5ml brown	neoLab Migge GmbH, Heidelberg, Germany
10	Nitrile gloves	Abena A/Sm Aabenraa, Denmark

11	Non-tissue culture treated plates (6-/24-well)	BD Biosciences, Franklin Lakes, USA
12	NuncTM Cell culture flask (80cm2)	Thermo Fisher Scientific, Waltham, USA
13	Parafilm M laboratory film	Pechiney Plastic Packaging, Chicago, USA
14	Pipet tips (10/20/300/1250 μl)	Sarstedt AG & Co., Nümbrecht, Germany
15	Screw Cap Micro Tubes	Sarstedt AG & Co., Nümbrecht, Germany
16	Sealing foil (ELISA)	Alpha Laboratories, Hampshire, UK
17	Serological Pipets (5ml, 10ml, 25ml, 50ml)	Sarstedt AG & Co., Nuümbrecht, Germany
18	Stericup/Steritop 0.22 μm filters	Merck KGaA, Darmstadt, Germany
19	Syringe filters (0.2, 0.45 μm)	TPP Techno Plastic Products AG, Trasadingen, Schweiz
20	Tissue culture-treated plates (48-well)	BD Biosciences, Franklin Lakes, USA
21	Tissue culture-treated plates (6-/12-/24-well, round/flat bottom 96-well)	TPP Techno Plastic Products AG, Trasadingen, Schweiz

Table 2.2: Consumables

2.1.3 Chemicals and Reagents

#	**Chemical, Reagent**	**Company**
1	AIM V™	Thermo Fisher Scientific, Waltham, USA
2	Bovine Serum Albumine (BSA)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany

3	Cyclosporin A	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
4	DEPC H_2O	Thermo Fisher Scientific, Waltham, USA
5	Dulbecco's Modified Eagle Medium (DMEM)	Thermo Fisher Scientific, Waltham, USA
6	Dimethylformamide (DMF)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
7	Dimethyl sulfoxide (DMSO)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
8	Ethanol	Merck KGaA, Darmstadt, Germany
9	Fetal calf serum (FCS)	Thermo Fisher Scientific, Waltham, USA
10	Ficoll	Biochrom GmbH, Berlin, Germany
11	Ionomycin	Merck KGaA, Darmstadt, Germany
12	Isopropanol	Merck KGaA, Darmstadt, Germany
13	L-Glutamine	Thermo Fisher Scientific, Waltham, USA
14	Milk powder	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
15	Non-essential amio acids (NEAA)	Thermo Fisher Scientific, Waltham, USA
16	Opti-MEM® I	Thermo Fisher Scientific, Waltham, USA
17	Paraformaldehyde (PFA)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
18	PBS (Gibco)	Thermo Fisher Scientific, Waltham, USA
19	PBS powder without Ca^{2+}, Mg^{2+}	Merck KGaA, Darmstadt, Germany
20	Penicilline/Streptomycin	Thermo Fisher scientific, Waltham, USA

21	Phorbol 12-myristate 13-acetate (PMA)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
22	Prostaglandine E2 (PGE2)	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
23	Protamine Sulfate	MP Biomedicals GmbH, Illkirch, France
24	RPMI-1640	Thermo Fisher Scientific, Waltham, USA
25	Sodium acetate (C2H3NaO2)	Merck KGaA, Darmstadt, Germany
26	Sodium azide (NaN3)	Merck KGaA, Darmstadt, Germany
27	Sodium Pyruvate	Thermo Fisher Scientific, Waltham, USA
28	Streptavidin-HRP	Mabtech AB, Nacka Strand, Sweden
29	Sulfuric acid	Carl Roth GmbH + Co. KG, Karlsruhe, Germany
30	TRIzol reagent	Thermo Fisher scientific, Waltham, USA
31	Trypane blue	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
32	Trypsine/EDTA	Thermo Fisher Scientific, Waltham, USA
33	Tween 20	Sigma-Aldrich Chemie GmbH, Taufkirchen, Germany
34	VLE-RPMI 1640	Biochrom GmbH, Berlin, Germany

Table 2.3: Chemicals and Reagents

2.1.4 Buffers & media

#	Solution/buffer	used for	Ingredients
1	Acetate buffer	ELISpot	46.9 $ml\ H_2O$ + 4.6 $ml\ C_2H_4O_2$ (0.2 M) + 11 $ml\ C_2H_3NaO_2$ (0.2 M)

2	AEC buffer	ELISpot	$500\ \mu l$ AEC solution + $9.5\ ml$ Acetate buffer, filtered ($0.45\ \mu m$)
3	AEC solution	ELISpot	$20\ \mu l$ AEC chromogen ($\widehat{=}$ 1 drop) in $1\ ml$ of AEC substrate
4	Blocking solution	ELISA	PBS + $1\%\ (m/v)$ milk powder
5	ELISA coating buffer	ELISA	H_2O + $0.1\ mol/l\ NaHCO_3$ + $0.03\ mol/l$ Na_2CO_3, pH = 9.5
6	HRP-complex solution	ELISpot	$10\ ml$ PBS + $50\ \mu l$ von Strp. / HRP + $50\ \mu l\ \Delta FCS$
7	Washing buffer	ELISpot, ELISA	PBS + $0.05\%\ (v/v)$ Tween 20
8	ΔFCS	various	FCS, inactivated for 20 min at 58 °C
9	ΔHS	various	HS, inactivated for 20 min at 58 °C

Table 2.4: Composition of solutions and buffer

#	Medium / buffer	Ingredients
1	AIM-V	AIM-V (Thermo Fisher Scientific), no supplements
2	cRPMI	RPMI supplemented with $10\%\ \Delta FCS$, $10\ mM$ non-essential amino acids, $1\ mM$ sodium pyruvate, $2\ mM$ L-Glutamine, $100\ U/ml$ Penicillin and $100\ \mu g/ml$ Streptomycin
3	Freezing medium	$90\%\ \Delta FCS$ + $10\%\ DMSO$
4	T-cell medium (TCM)	RPMI 1640 supplemented with 5% v/v ΔFCS, $5\%\ \Delta HS$, $10\ mM$ non-essential amino acids, $1\ mM$ sodium pyruvate, $2mM$ L-Glutamine, $100\ U/ml$ Penicillin, 100 $\mu g/ml$ Streptomycin, $10\ mM$ HEPES buffer and $16.6\ \mu g/ml$ Gentamycin

Table 2.5: Composition of media

2.1.5 Kits & Antibodies

#	Kit	Application	Company
1	BD OptEIA™Human IL-2 ELISA Set	Measurement of cytokines in supernatant	BD Biosciences, Franklin Lakes, USA
2	BD OptEIA™Human IFN-γ ELISA Set	Measurement of cytokines in supernatant	BD Biosciences, Franklin Lakes, USA
3	BD OptEIA™TMB Substrate Reagent Set	Measurement of cytokines in supernatant	BD Biosciences, Franklin Lakes, USA
4	Venor GeM OneStep mycoplasma detection kit	Testing cell lines for infection with mycoplasma	Minerva Biolabs GmbH, Berlin, Germany

Table 2.6: Kits

#	Antibody	Clone	Conjugation	Company
1	anti-human IFN-γ mAb (coating)	1-D1K	-	Mabtech AB, Nacka Strand, Sweden
2	anti-human IFN-γ mAb (capture)	7-B6-1	biotinylated	Mabtech AB, Nacka Strand, Sweden

Table 2.7: Antibodies

2.1.6 Cytokines

#	Cytokine	Company
1	OKT-3	provided by Elisabeth Kremmer
2	Poly-I:C	InvivoGen, San Diego, USA
3	recombinant human GM-CSF	PeproTech, London, UK
4	recombinant human IFN-g	PeproTech, London, UK
5	recombinant human IL-1b	PeproTech, London, UK
6	recombinant human IL-2	PeproTech, London, UK
7	recombinant human IL-4	PeproTech, London, UK
8	recombinant human IL-7	PeproTech, London, UK
9	recombinant human IL-15	PeproTech, London, UK
10	recombinant human TNF-a	PeproTech, London, UK

Table 2.8: Cytokines

2.1.7 Peptide order

#	Patient	Seq_ID	Order-No. (Dgpeptides)	Sequence	Molar mass [g/mol]
1	ImmuNEO-01	IN_01_a	S-4071	ALSGHLETL	940,07
2		IN_01_b	S-4072	DAAGRNSW	875,9
3		IN_01_c	S-4073	GMGSESKASF	1000,1
4		IN_01_d	S-4074	KGDSPQVKLKY	1262,48
5		IN_01_e	S-4075	KKGGLIGS	758,92

6		IN_01_f	S-4076	LEAKGQAL	828,97
7		IN_01_g	S-4077	LEHGGAIMA	898,06
8		IN_01_h	S-4078	LLGSAVHE	824,92
9		IN_01_i	S-4079	SHLYSDPG	874,89
10		IN_01*wt_b	S-4080	DAARRNSW	975,01
11		IN_01_2	/	FLAKKPSAV	960,17
12		IN_01*wt_1	/	FLAKKSSAV	950,13
13		IN_01_1	/	ALAAVVTEV	872,01
14	ImmuNEO-04	IN_04_a	S-4181	AGPGNRVL	782,88
15		IN_04_b	S-4182	AGVVLGGL	684,82
16		IN_04_c	S-4183	CVYKNPVI	935,14
17		IN_04_d	/	FFTLISVSF	/
18		IN_04_e	/	FLALFWITI	/
19		IN_04_f	S-4186	FLLLLLKNF	1120,42
20		IN_04_g	S-4187	GAGALLCTHL	955,13
21		IN_04_h	S-4188	GLAATFASL	849,97
22		IN_04_i	S-4189	GSPGGPVSI	769,84
23		IN_04_j	S-4190	HVGGAGLEHL	989,08
24		IN_04_k	S-4191	KTKEMSNNVK	1178,35
25		IN_04_l	S-4192	LGGTGASF	708,75
26		IN_04_m	S-4193	NTLMSLSDM	1011,17
27		IN_04_n	S-4194	QKRLYYQLFFNCSWY	2059,34
28		IN_04_o	S-4195	SLPQNLLYL	1060,24
29		IN_04_p	S-4196	SYLSNISY	946

30		IN_04_q	S-4197	THIDAGRF	915,99
31		IN_04_r	S-4198	TSLAANTF	823,89
32		IN_04_s	S-4199	TVHSTSIAF	962,05
33		IN_04_h	S-4200	GLTATFASL	879,99
34	ImmuNEO-05	IN_05_b	S-4201	ASQTAGIAGVR	1030,16
35		IN_05_c	S-4202	DIFSRISQR	1121,24
36		IN_05_d	S-4203	ETNKSLLKR	1088,28
37		IN_05_e	S-4204	FLSLADHAT	974,09
38	ImmuNEO-09	IN_09_a	S-4205	YHLMPFRQHCWQSL	1846,14
39	ImmuNEO-11	IN_11_a	S-4206	AAAAPARGL	796,91
40		IN_11_b	S-4207	ARETLLETL	1045,18
41		IN_11_c	S-4208	ETSAPASSL	861,89
42		IN_11_d	S-4209	GTPSSTTL	762,8
43		IN_11_e	S-4210	ISAAELHHV	976,08
44		IN_11_f	S-4211	LNITHGILY	1043,21
45		IN_11_g	S-4212	LNLREKKNK	1142,35
46		IN_11_h	S-4213	RLQDAVPV	897,03
47		IN_11_i	S-4214	SAAELHHV	862,92
48		IN_11_j	S-4215	SRAAAAPAR	869,96
49		IN_11_e	S-4216	ISAAELRHV	995,13
50	ImmuNEO-15	IN_15_a	S-4225	RVVHVSTSQK	1140,29

51	ImmuNEO-19	IN_19_a	S-4081	DQATCLRSTKFTIY	1646,86
52		IN_19_b	S-4082	FFQDKAWFY	1251,38
53		IN_19_c	S-4083	GRPGTRPAL	924,05
54		IN_19_d	S-4084	GWGVAGTM	777,88
55		IN_19_e	S-4085	ITRGQEFE	979,07
56		IN_19_f	S-4086	LLEAGRLR	927,12
57		IN_19_g	S-4087	PTDAELMS	862,96
58		IN_19_h	S-4088	SESNVDRLM	1050,16
59		IN_19_i	S-4089	STLVLDEFKR	1207,4
60		IN_19_j	S-4090	TLGGWGGQDLR	1159,28
61		IN_19_k	S-4091	TNLGFSKK	894,05
62		IN_19_l	S-4092	VASISLTK	817,99
63	diverse	IN_09_b	S-4094	VVHVSTSQK	984,13
64	diverse	IN_05_c	S-4093	QLRASGQLK	1000,17
65	diverse	IN_09_b_WT	S-4096	VVHLSTSQK	998,16
66	diverse	IN_05_c_WT	S-4095	HLRASGQLK	1009,19

Table 2.9: Peptides order from DGpeptides Co., Ltd, Hangzhou city, Zhejiang province, China.

2.2 Computational Methods: Dataset Analysis

2.2.1 Genetic variants

All subsequent considered peptides, in the following described as proteogenomic neoantigen candidates (NACs), and their immunological testings originate from the dataset anal-

#	R-Package	Version	Description
1	base	*4.2.1*	The R base package
2	dplyr	*1.0.9*	A grammar of data manipulation
3	ggHighlight	*0.4.0*	Highlight lines and points
4	ggplot2	*3.4.0*	Create Elegant Data Visualization
5	ggrepel	*0.9.1*	Automatically position non-overlapping text
6	graphics	*4.2.1*	The R graphics package
7	gridExtra	*2.3*	Miscellaneous Functions for "Grid"
8	tidyverse	*1.3.2*	Tidy messy data
9	Eulerr	*6.1.1*	Area-Proportional Euler and Venn Diagrams
10	readr	*2.1.2*	Read rectangular data

Table 2.11: Used software packages for R Studio.

ysis of genetic variants. These data contain DNA mutations found on exome level, as well as RNA alterations found on RNA level (transcriptome). Mutation calling was performed by Sebastian Lange. For details on the method, see (Lange et al., 2020; Kim et al., 2018; Benjamin et al., 2019).

For further analysis of the genetic variants, all corresponding tsv-files for all patients, forming the dataset with all identified variants, were imported to "R" and merged into one data frame. With this, the corresponding variable "Metastasis" was assigned via file-name insertion. See Appendix A, 2.1 for details.

2.2.1.1 R packages

For all data set analysis RStudio 2022.07.1 Build 554 was used on macOS 12.6.1. The relevant R-packages are listed in table 2.11.

2.2.1.2 Data overview

The DNA data set contained 40 tumor samples from 32 patients. The RNA data set contained 32 tumor samples from 26 patients.

The different correlation and cleanup procedures can be reproduced using the corre-

sponding R-script (see Appendix A, 2.1). For the visualization of overlapping sets of data (i.e., the overlap of genomic variants of two metastases), the R-package *eulerr* (see: `https://rdrr.io/cran/eulerr/`) was used.

2.2.1.3 Tumor variant frequency

Certain parameters from **WES!** (**WES!**) and RNA sequencing (RNA-Seq) were used to assess particular quality control aspects of the dataset. The number of mutated and non-mutated reads for tumor and normal samples, respectively, could be used as markers to evaluate the validity of each identified variant.

For this purpose, the tumor **VF!** (**VF!**), that is a quantity describing the mutation coverage at a particular locus, was calculated:

$$VF_{tumor} = \frac{\#\, reads_{mutated}}{\#\, reads_{mutated} + \#\, reads_{WT}} = \frac{AD_{tumor}}{AD_{tumor} + RD_{tumor}} \tag{2.1}$$

It is essential to notice that the significance of the tumor **VF!** depends on the sequencing depths and the number of total reads in the tumor. In order to account for this fact, several filtering criteria are discussed in the next step.

2.2.1.4 Filtering

For all further characterization of the mutational landscape, the dataset was modified such that every observation referred to a unique variant. This unique variant was defined by the following variables: chromosome number (CHROM),position of the base pair (POS),reference nucleotide (REF),altered nucleotide (ALT).

In the next step, all genetic variants were split up into two groups based on their **VF!**, their mutated coverage (mutated reads), and their total coverage (total reads). The following values were applied as thresholds (TH):

$$VF_{tumor} \geq 0.05 \tag{2.2}$$

$$AD_{tumor} \geq 3 \tag{2.3}$$

$$AD_{tumor} + RD_{tumor} \geq 5 \tag{2.4}$$

Preliminarily, all variants in the dataset that exhibited more than one mutated read in the normal control (RD_{normal}) tissue had been rejected.

2.2.2 Neoantigen candidates

Before any experimental validation of neoantigens, the dataset of possible NACs had to be analyzed. Input data for the used pipeline were two .tsv-files (originating from pFind (Chi et al., 2015) and from PROSIT (Gessulat et al., 2019) raw data), that were imported to R.

The contained columns for pFind were *"patientID", "CHROM", "POS", "Seq", "SeqMarked", "SeqGroup", "multiGene", "multiEntity", "multiPatient", "gene", "calledBy", "comment", "TrueHit", "nReps", "nFound", "mutationType", "transcriptTypes", "geneBiotype", "transcriptBiotype", "EFFECT", "scoreMS", "TumorAD.Mutect2", "TumorRD.Mutect2", "NormalAD.Mutect2", "NormalRD.Mutect2", "TumorAD.StrelkaRNA", "TumorRD.StrelkaRNA", "NormalAD.StrelkaRNA", "NormalRD.StrelkaRNA", "header"*.
The contained columns for PROSIT were *"patientID", "Seq", "SeqMarked", "gene", "TrueHit", "calledBy", "nReps", "nFound", "CHROM", "POS", "mutationType", "transcriptTypes", "geneBiotype", "transcriptBiotype", "EFFECT", "PostErrorProb", "TumorAD.Mutect2", "TumorRD.Mutect2", "NormalAD.Mutect2", "NormalRD.Mutect2", "TumorAD.StrelkaRNA", "TumorRD.StrelkaRNA", "NormalAD.StrelkaRNA", "NormalRD.StrelkaRNA", "header", "REF", "ALT"*.

The two created data frames were merged and annotated by additional reference data (Master_ID/Patient_ID, HLA-types, and tumor entity data). Next, tumor **VF!**s for Mutect2 and Strelka were calculated, and .csv-files were generated for predictions with netMHC and MHCflurry (cp. Section 2.2.3). After merging the data frame with the obtained rank and binding affinity predictions, the best binding alleles were extracted, and duplicates were eliminated patient-wise.

Next, a FASTA file was generated to be used with basic local alignment search tool (BLAST) provided by the National Center for Biotechnology Information (NCBI) (Boratyn et al., 2013). On `https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome` the FASTA-file was sub-

#	True Hit	Case description	Procedure
1	yes	The nucleotide sequence encoding the emerged peptide contains the called mutation.	Inclusion
2	no	The same ORF contains the called mutation and the peptide representing nucleotide sequence without any overlap.	Exclusion
3	maybe	Same as in case "no" but the mutation is either a frameshift (FS) mutation being upstream (US) of the peptide sequence or a splice donor/splice acceptor mutation or the peptide sequence is fully intronic.	Individual decision based on blat

Table 2.13: Distinction of different cases arising from mutation calling.

mitted (Algorithm: *blastp*; protein-protein BLAST) and the received *Alignment-Hit-Table* was imported to R. After processing and merging to the peptide data, sequences with more than two BLAST-hits were filtered out. Sequences that produced one or two hits in the BLAST search were checked manually by individual database research.

To provide information about the peptide underlying mutation, the nucleotide sequence of the peptide was mapped to the associated segment of the genome (to blat) with the genome browser supplied by the **UCSC!** (**UCSC!**) (Kent, 2002; Kuhn et al., 2013). Depending on the *"TrueHit"*-classification (see table 2.13) a priori three cases were distinguished: For all cases classified as "maybe", an individual decision for each peptide was made considering the present peptide-mutation context found by blatting. Here the crucial criteria were intron inclusion, the relative position of mutation, the position of peptide, mutation effects on the start/stop codon, frame of peptide, and peptide aberrance from the assigned sequence of WT-peptide. Additionally, it was checked if the mutation-induced nucleotide sequence could generically be found on another (WT-)gene or chromosome ("multimapping"). A detailed scheme for the classification of NACs based on the assessment of the criteria mentioned above can be found in Appendix C. For all NACs, a unique sequence ID (*Seq_ID*) was assigned that was used to create a list for peptide orders.

Finally, the reference (REF) and the alternative (ALT) sequence at the mutation site were extracted from the header, and a new data frame with WT peptides was generated. This was processed similarly as the data frame for NACs, with the corresponding binding affinities of the non-mutated WT-peptides.

#	Method	Strong Binder ("SB")	Weak binder ("WB")	No binder
1	Percentile rank based	$< 0.5\%$	$< 2.0\%$	$> 2.0\%$
2	Binding affinity based	$K_d < 50nM$	$K_d < 500nM$	$K_d > 500nM$

Table 2.15: Thresholds for affinity- and rank-based peptide selection.

The respective R-script can be found in Appendix A, 2.3.

2.2.3 Prediction of peptide-MHC class I binding affinities

One of the major problems in neoepitope discovery deals with the effectiveness of antigen presentation and, based on this, with the binding between peptide and the patient's MHC class I molecules. This section describes the applied in silico methods to estimate this binding strength.

2.2.3.1 Technical background

In order to assess the quality of this binding, there are several neural network-based methods, such as netMHC or MHCflurry, that can estimate the binding affinity of an epitope, based on typically 50-100 experimentally validated peptide-binding affinity measurements. These predictions can either be performed by an "allele-specific" approach (Andreatta & Nielsen, 2016), whereby separate predictors are trained for each MHC-I allele individually (O'Donnell et al., 2018) or by a "pan-allelic" approach. In the former case, the model is closed, meaning that only the peptide sequence of interest is taken as an input. The model using "pan allelic" architecture, integrates both inputs, the peptide sequence, and a representation of the MHC allele (Nielsen & Andreatta, 2016a). Considering the vast number of different MHC-I types, a pan-allele approach represents a feasible option and may be advantageous for rare HLA types.

2.2.3.2 Data acquisition procedure and threshold selection

To prioritize potential strong binding peptides that generally are associated with a higher probability of antigen presentation (Gfeller & Bassani-Sternberg, 2018) and thus even-

tually lead to higher immunogenicity, all NACs were assessed utilizing two different prediction algorithms. Here, all possible pairings of NAC and disposable HLA-alleles (HLA-alleles that the corresponding patients inherited) were considered. The best binding HLA-allele for each patient/NAC group was determined with both prediction methods. The value of the corresponding nano-molar affinity and the MHC-associated percentile rank was used to quantify the binding affinity. Here, the percentile rank represents the ranked binding affinity compared to a large set of random natural peptides (Andreatta & Nielsen, 2016). Fixed thresholds for strong and weak binders were chosen, according to the recommendations of (Moutaftsi et al., 2006) and (Zhao & Sher, 2018) for both percentile rank and nano-molar binding affinity method, respectively. These are displayed in table 2.15. A prioritization of NACs was then made based on the above-raised values.

To assess the MHC class I binding affinity of potential NACs (cp. Section 2.2.2), two different tools for binding affinity prediction were used (see table 2.17). The technical application of both tools is described in the following.

The prerequisite for binding affinity prediction was the result of the corresponding HLA typing, which was done by AG Rad with the help of xHLA (Xie et al., 2017), BWAKit (Li, 2013) and OptiType (Szolek et al., 2014).

2.2.3.3 netMHC 4.0

Setup A Linux-based version of the software *netMHC 4.0* provided by the *Department of Health Technology* of the *Technical University of Denmark* was requested for academic use at `https://services.healthtech.dtu.dk/cgi-bin/sw_request`. The tool was installed according to instructions on a dedicated 1 CPU, 2GB RAM server with *CentOS 7-7.1908*. As a requirement for the installation, the package *tcsh.x86_64EMZ9* had to be installed.

Input A FASTA file with the corresponding peptide sequences was generated for each patient with *R*. The associated three to six MHC alleles (HLA-A, HLA-B, and HLA-C) per patient were filtered according to their availability in the set of trained netMHC predictors and subsequently exported as .csv-file with *R*.

#	Software Package	Version	Model	Predictor Type
1	MHCflurry	*2.0.0*	models_class1_presentation	allele-specific
2	MHCflurry	*2.0.0*	models_class1_pan	pan-allele
3	netMHC	*4.0a*	-	allele-specific

Table 2.17: Used software releases and models for binding affinity prediction.

Implementation A Python script was used (see Appendix A, 1.1) to pass the sequences with the appropriate alleles to *netMHC 4.0*. The obtained .xls (.tsv) files were imported to *R* and integrated with the existing peptide data.

2.2.3.4 MHCflurry

Setup The open-source software package *MHCflurry* was installed with Python package installer (pip) *20.0.2* on a macOS *10.12.6* together with the required packages.

Input A CSV file, with the columns *peptide* and *allele*, was used as input for *MHCflurry*. The sequences in column *peptide* were in 1-Letter notation, *allele* contained the corresponding HLA types. The additional columns *Patient_ID, Master_ID, Seq_ID* and *allele_nr* were only used for data integration with R.

Implementation A Python script was used to loop through all input files and to execute MHCflurry for each patient-specific input file. The code can be found in Appendix A, 1.2. The resulting .csv files were imported to *R* and integrated with the existing peptide data.

2.3 Cell biological Methods

2.3.1 Cell-Culture

Until May 2019, thawing, freezing, and cultivation of cells was performed in the laboratory unit in building 549 (Schneckenburgerstr. 6). All later cell-culture-related work was

realized in laboratories in building 522 (Einsteinstr. 25). Work with primary cells and cell lines, as well as EBV-transformed cells were done in agreement with S1 and S2 safety guidelines and under sterile conditions.

2.3.1.1 Thawing, counting and freezing of cells

After extraction of cell-media suspension containing cryotubes from the liquid nitrogen storage and subsequent cooled transport, cells were quickly thawed using a 37 °C water bath until being partly melted. By adding small amounts of RPMI, they were then transferred to a falcon with pre-tempered media at the ratio of 1:10 for the successive washing step. Herefore, the cells were spun down for 5 min 500 g, and the supernatant was poured off and replaced by 1 ml of fresh media.

For consecutive counting, 10 µl of cell suspension was diluted in Trypane blue according to the expected cell concentration and pipetted on a Neubauer counting chamber, which was then analyzed under the light microscope, such that all alive single cells could be counted in all four quadrants. The corresponding cell concentration could then be calculated by:

$$c_{cells}\,[1/ml] = \frac{N_{cells,\,counted}}{N_{quadrants,\,counted}} \cdot d \cdot 10^4 \tag{2.5}$$

where d is the dilution factor of the cell suspension in Trypane blue.

For cryo-preservation, cells that were beneficial for future applications were washed, resuspended in a freezing medium, and transferred to cryotubes with an aliquot size of 1 ml. Using a pre-cooled *Nalgene* Cryo 1 °C Freezing Container, the cryotubes were cooled down to -80 °C and then relocated to the liquid nitrogen tank the next day for long-term storage.

2.3.1.2 Cultivation of cell lines

Through known cell concentrations, the cell-media suspension could be diluted with the corresponding amount of media for further cultivation. For cell lines, freshly prepared complete RPMI (cRPMI) was used, and cells were split up twice a week or if a sufficiently high cell density was assumed, according to the color of the media. Cells were stored at 37

°C, and their growth was frequently controlled and evaluated under the light microscope. Primary cells were transferred to AIM V according to the protocol for in-vitro stimulation of T cells (see Section 2.3.2).

2.3.1.3 Isolation of primary cells

To attain sufficient amounts of peripheral blood mononuclear cells (PBMCs) for further in-vitro stimulation, whole blood (or leukapheresis products) had to be separated into their components. Ficoll gradient density centrifugation (Noble & Cutts, 1967) was used for this. 35 ml of RPMI-diluted whole blood (ratio 1-2:1) was slowly poured into a 50 ml falcon containing 15 ml of previously prepared Ficoll solution, such that the boundary layer between both matters was maintained. After 25 minutes of centrifugation at 880g, using the option for reduced acceleration and deceleration, the leukocyte layer was carefully removed with a serological pipet. Last, the obtained suspension was washed with RPMI twice for further use in stimulation assays or other applications.

2.3.1.4 Generation of EBV-transformed lymphoblastoid cell lines (LCL)

Supernatant from Epstein-Barr virus (EBV) was used to induce in-vitro transformation of B-cells to obtain immortalized lymphoblastoid cell lines (LCLs) (Anderson & Gusella, 1984) for later antigen presentation. Initially, semi-adherent growing B95-8 cells had to be seeded and expanded in cRPMI until reaching a concentration of $0,5 - 1 \cdot 10^6$ cells. After being stimulated with PMA (c_{PMA}=20 ng/ml) for one hour at 37 °C, cells were washed and taken back to culture for further three days. The harvested supernatant was then purified with 0,45 µm sterile filters and was stored in aliquots of 1 ml at -80 °C until use.

Next, 5 Mio freshly isolated or thawed PBMCs were resuspended in 1 ml cRPMI and incubated with 1 ml EBV supernatant for two hours at 37 °C. After adding 1,5 µl of Cyclosporine A ($c_{cyc\text{-}A}$=2 mg/ml) and 1 ml of additional cRPMI, the infected cells were transferred to two T25 flasks. They were then further incubated at 37 °C.

Once a week, 1 ml of medium was changed with fresh cRPMI until after 3-5 weeks, macroscopically visible clusters indicated a successful transformation. Further expansion was performed until reaching sufficient cell numbers. Cells were then cryopreserved

until use.

2.3.2 In-vitro stimulation of T cells

To check for antigen-specific T-cell responses in primary cells, a modified version of an accelerated co-cultured dendritic cell (acDC) assay (Martinuzzi et al., 2011) was performed. Here T cells were stimulated, expanded, and co-cultured with antigen-presenting cells (APCs) before measuring their interferon-γ (IFN-γ) production with ELISpot (Ranieri et al., 2014).

2.3.2.1 T-cell stimulation and DC culture

Patient PBMCs were thawed, washed, counted (cp. Section 2.3.1.1) and cultivated in AIM V, in a 96-well plate (flat bottom) supplemented with 0.1 ng/µl granulocyte-macrophage colony-stimulating factor (GM-CSF) and 0.1 ng/µl interleukin 4 (IL-4). Depending on patient-related sample availability, the applied cell concentrations varied between 10^5 and $6 \cdot 10^5$ cells per well.

After incubation at 37 °C for 24 hours peptide (c_{end}=1 ng/µl; solved in DMSO) was added along with interleukin 7 (IL-7) (c_{end}=0.5 ng/ml), **TNF-a!** (**TNF-a!**) (c_{end}=50 ng/ml) and interleukin-1β (IL-1β) (c_{end}=10 ng/ml). tetradecanoyl phorbol acetate (PMA) (c_{end}=1 ng/µl) and Ionomycin (c_{end}=2 ng/µl) were added to wells selected for positive control. Instead of peptide solution, 1 µl of DMSO was added to wells selected for negative control. Cell transfer to previously coated ELISpot plates was performed after 24 hours of incubation at 37 °C and washing with AIM V. The ELISpot assay is described in Section 2.3.2.2.

Afterward, T cells were re-transferred to a 96-well plate (round bottom), washed and cultivated in TCM supplemented with c_{end}=5 ng/ml IL-7 and c_{end}=5 ng/ml IL-15 for 10-12 days. During the expansion of T cells, IL-7 and IL-15 were added every two days, and the medium was changed according to the cell density as indicated by the color. To ensure an appropriate cell concentration, all wells were checked under the light microscope twice a week, and cells were eventually transferred to a suitable larger well plate.

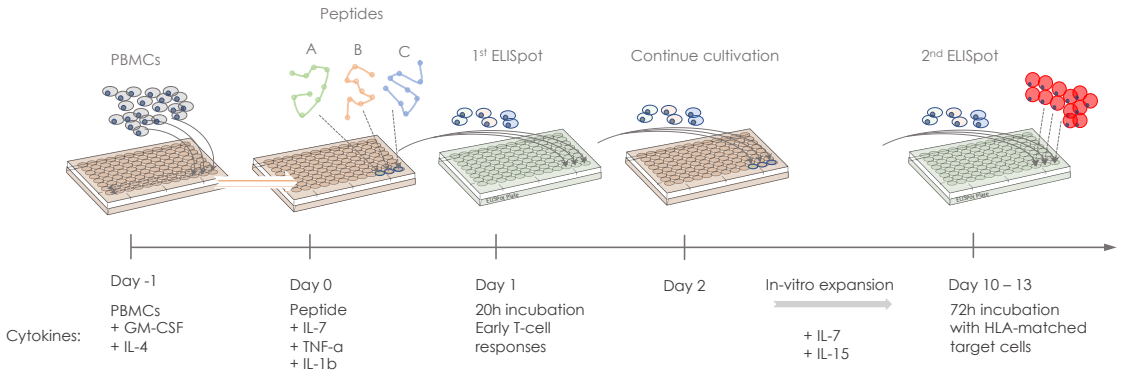


Figure 2.1: **Schematic representation of T-cell stimulation, expansion, and corresponding analysis with ELISpot.** Two ELISpot analyses were performed, one 24 hours after stimulation with peptides and cytokines, the other at day 10-13 after expansion of T cells with IL-7 and IL-15. The second ELISpot measures IFN-γ secretion caused by co-incubation with target cells.

2.3.2.2 Immunogenicity assessment with ELISpot

At day 0, capture antibody solution ($c_{AB\ 1\text{-}D1K}$=10 ng/µl; solved in PBS) was freshly prepared, corresponding ELISpot plates were coated with 50 µl per well and incubated at 4 °C overnight. Two hours before use, the antibody was discarded, and the ELISpot plate was washed four times with PBS (200 µl), each time leaving it to incubate at RT for 10 min. Afterward, all wells were blocked with 150 µl TCM for at least 45 minutes at 37 °C.

To detect early T-cell reactivities (day 1), cells were transferred to the previously coated ELISpot plate directly after discarding the block and incubated for 20 hours at 37 °C.

For detection of late T-cell responses (after expansion; day 10-13), an equal number (10000-20000) of preliminary peptide-pulsed target cells (cp. Section 2.3.1.4) was co-incubated with the T cells for 72 hours at 37°C. Here, all conditions (pulsed, WT-pulsed, unpulsed, positive control, negative control) were placed on the same ELISpot plate. After incubation, the supernatant was taken off and stored at -20 °C for later analysis. The cells were then either transferred back to a 96-well plate (round bottom) for further cultivation, washing and freezing, or they were discarded.

For the development of the assay, the ELISpot plates were washed six times with washing buffer, beating the plates to dry between each washing step. Next, IFN-γ antibody 7-B6-1-Biotin was diluted in PBS with 0,5% BSA (c_{AB}=2 ng/µl) and 100 µl were added to each well for subsequent incubation at RT for 2 hours. After repeating the washing

procedure, 100 µl of streptavidin horseradish peroxidase (Strep-HRP) complex solution was added to each well for a further 90 minutes of incubation in the dark and at RT. The consecutively performed washing step was done twice with washing buffer and then twice with PBS only.

To visualize bound components, 100 µl of AEC solution was added to each well, and the ELISpot plate was then quickly stored in the dark for 5-15 min of incubation at RT. When spots were clearly visible (positive control), the reaction was stopped with abundant deionized water. To prevent smearing of spots, all plates were directly dried out for 30 min with the help of the airflow in the fume hood and then stored in the dark until further analysis. Read-out was done within three days on an ImmunoSpot S6 Ultra-V Analyzer using Immunospot software 5.4.0.1 (CTL-Europe).

The detailed protocol can be found in Appendix C, 3.2.2.

2.3.2.3 Cloning by limiting dilution

To detect reactivities in the case of clonal T cells and possible TCR identification, single clones had to be extracted from the previously expanded T-cell culture by dilution to a final concentration of $\sim$ one cell / well. These single T cells were co-cultured in TCM with previously γ-irradiated (30 Gy) feeder cells at a concentration of 50000 feeder cells per well. Additionally, the medium was supplemented with 50 U/ml IL-2, 5 ng/ml IL-7, 5 ng/ml IL-15, and 30 ng/ml OKT-3. To avoid wrong cell concentrations due to possible counting errors, the dilution of T cells was performed such that two different final T-cell concentrations (c_1=1 cell/well, c_5=5 cells/well) were realized. 200 µl of each dilution was plated on four 96-well plates.

For cultivation, IL-2 was added every three days to a concentration of 50 U/ml, and IL-7 and IL-15 were added once a week to a concentration of 5 ng/ml. In the case of observable clones after 15 to 20 days, T cells were analyzed for specific responses with ELISpot (cp. Section 2.3.2).

2.3.2.4 ELISA

For the detection of cytokine release during different incubation procedures (early T-cell responses on ELISpot plates or co-cultures with target cells) *BD OptEIATM Human IFN-γ, IL-2* or ***TNF-a!*** *ELISA Set* was used following the instructions of the manufacturer. 150 µl of cell culture supernatant was gained directly after incubation and was stored at -20 °C. Coating of a 96-well ELISA plate was done one day before analysis. Therefore 50 μl of the corresponding capture antibody (IFN-γ, IL-2 or **TNF-a!**) diluted in ELISA coating buffer (1:250) had to be added to each well carefully, avoiding bubbles. Subsequently, a sealing foil was put on the plate and incubated at four °C overnight.

After washing the plate three times with washing buffer, 200 µl of a 1% m/v milk powder in PBS solution was added to each well, and the plate was incubated at RT for one hour for blocking. In the meanwhile, the supernatant, as well as the ELISA standard, were thawed. Since the ELISA standard stock concentration may vary for each lot, the corresponding lot information had to be considered for further dilution.

After dissolving stock solutions in incubation buffer (AIM V or TCM, respectively) to a concentration of c_{start}=1000 pg/ml for IFN-γ and c_{start}=500 pg/ml for IL-2 and **TNF-a!** a 1:1 titration series was performed in five microtubes, plus one microtube with buffer only:

$$
\begin{aligned}
&dilution\ (1) : buffer + ELISA\ standard\ \rightarrow c = c_{start} \\
&dilution\ (i) : 500\mu l\ buffer\ + 500\mu l\ dilution_{i-1};\ i = 2,..,6\ \rightarrow c = \frac{c_{start}}{2^{i-1}} \\
&dilution\ (7) : 500\mu l\ buffer\ \rightarrow c = 0
\end{aligned}
$$

The plates were washed three times with washing buffer, and the standard dilutions (1)-(7) (in duplicates) as well as the supernatant (50 µl per well each) were pipetted.

After 1 hour of incubation at RT, the plates were washed five times, and AB-enzyme conjugate solution was prepared diluting detection antibody (IFN-γ: 1:250, IL-2 and **TNF-a!**: 1:500) and HRP (1:250) in blocking solution. Subsequently, 50 µl of AB-enzyme conjugate solution was pipetted to each well, and plates were closed with the sealing foil and incubated for one hour at RT. The plates were washed seven times, and 100 µl of freshly prepared substrate solution (Substrate A + B from *BD OptEIATM TMB Substrate Reagent*

Set; mixed in a ratio of 1:1) for incubation was added in the dark to start the enzyme reaction.

After approx. 10-20 min, the wells containing the standard appeared from dark to light blue, which was used as a lead to stop the reaction by adding 50 ml of sulfuric acid.

For analysis, the optical density was measured with an absorbance at 450 nm and a reference of 570 nm with *Sunrise™absorbance reader*. The optical density was normalized to concentrations using the obtained calibration curve from the wells with the ELISA standard.

3 Results

3.1 Computational Analysis and Integration of Data

3.1.1 Overview of the genetic, proteogenomic and bioinformatical pipeline

To demonstrate the integration of this work into the context of the ImmuNeo project, a brief overview of the whole pipeline will be given (see Fig. 3.1). This includes different bioanalytical methods, such as NGS and MS, as well as other bioinformatical tools, like mutation calling and prediction of MHC class I binding affinities.

At the time of writing, the ImmuNeo MASTER cohort, which represents the foundation of this work, comprised 32 patients with different tumor entities. The corresponding list of tumor entities of all ImmuNeo patients can be found in Appendix B, 1. Data acquisition was performed in two initially independent strands that constitute the basis of the pipeline.

On the one hand, there is the data set containing the information on genetic variants from patients included in the ImmuNeo MASTER cohort or from ImmuNeo Plus samples. Here, analyzed at the DKFZ facility (Horak et al., 2021), two levels of genetic variants were acquired, exome data from **WES!** and **WGS!** (**WGS!**) as well as transcriptome data from RNA-Seq (*Genomics & Transcriptomics*). On the other hand, there is the proteogenomic data set containing information about the presented immunopeptidome, which was acquired by HLA class I peptide (pHLA) Immunoprecipitation with successive peptide purification (peptide elution) and MS/MS analysis (*Immunopeptidomics*).

As a first step, the raw data from **WES!**/**WGS!**, as well as from RNA-Seq, had to be analyzed by different tools to identify somatic single nucleotide variants (SNVs) and insertion-deletion mutations (indels). For this mutation calling, data from both sources

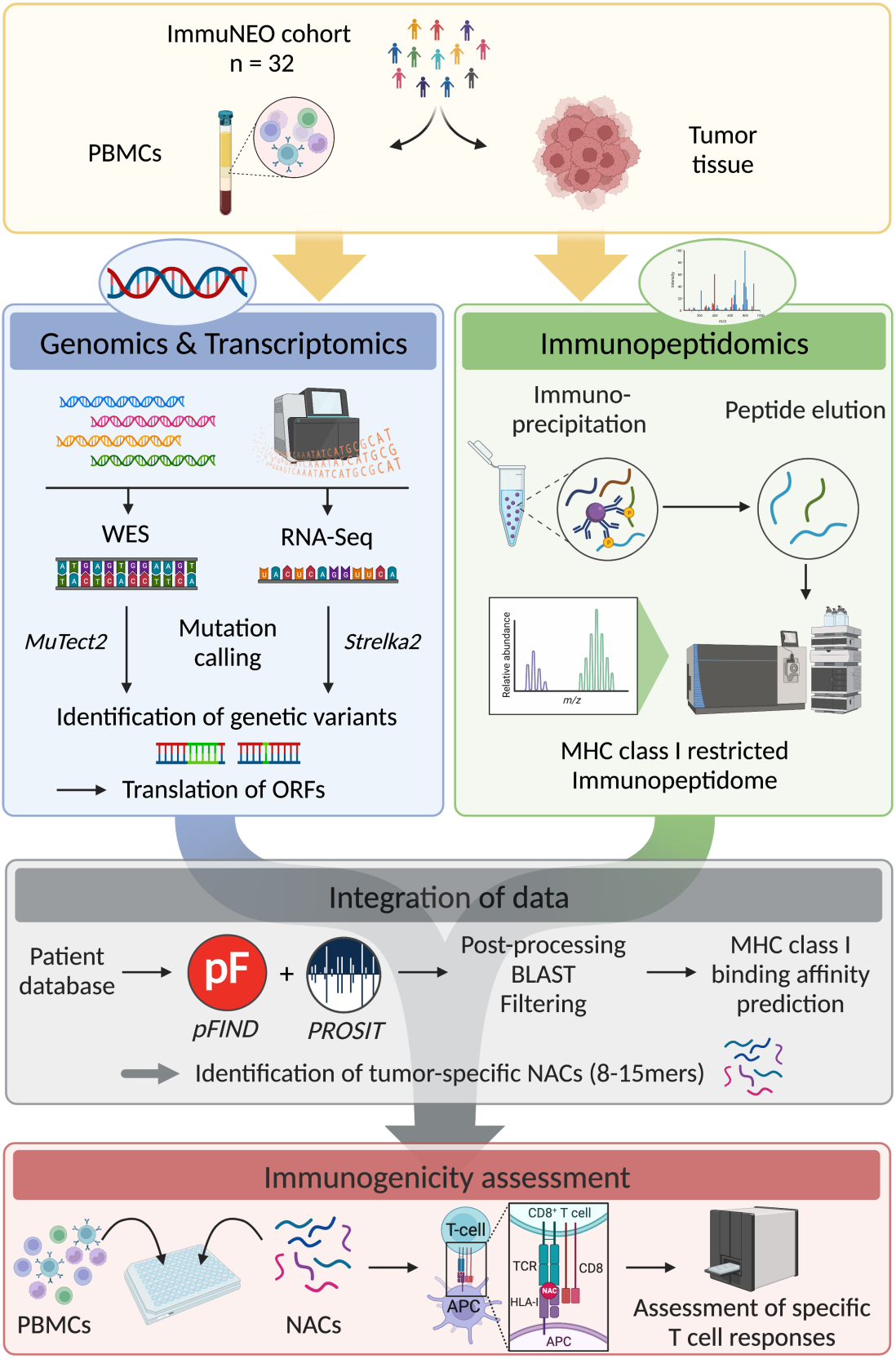


Figure 3.1: **Schematic representation of the pipeline. License: see List of Figures.**

were analyzed with the corresponding software *MuTect2* (Benjamin et al., 2019) and *Strelka2* (Kim et al., 2018; Lange et al., 2020), respectively. Furthermore, filtering of all identified variants for SNP was realized using the SNP database (dbSNP) (Sherry et al., 1999). In addition to the mutation data, the resulting **VCF!** (**VCF!**) file also contained variant frequencies that were integrated into filtering procedures.

For the subsequent **VCF!** translation, a **WT-DNA!** (**WT-DNA!**) sequence was obtained from *Ensemble92* protein database, and all relevant mutations were introduced. In a next step, all possible open reading frames (ORFs) were determined and translated to AAs. All new ORFs were predicted for non-coding regions or intron inclusions.

Next, MaxQuant (Cox & Mann, 2008; Tyanova et al., 2016), pFind (Chi et al., 2018), and PROSIT (Gessulat et al., 2019) were used to match the full length ORF (as FASTA format) to the spectral data from MS/MS such, that every possible tryptic peptide (8 to 15-mer) was considered. All the WT-peptides were filtered out using the Human Protein Atlas (Uhlen et al., 2017).

In the following post-processing, the resulting raw data were annotated (i.e., to obtain corresponding gene biotypes or MS tools for each peptide), and SNPs with an allele frequency (AF) greater than 1% were filtered out. Additional information from HLA-typing, as well as tumor **VF!** data, were annotated.

Subsequent predictions of MHC class I binding affinities were performed with the open-source tool MHCflurry (O'Donnell et al., 2018) and with netMHC, a formerly established proprietary tool for binding affinity estimation (Jurtz et al., 2017). Then, the best binding alleles were estimated for each peptide. Further details can be found in section 2.2.3. To check for already known peptide sequences in literature, a BLAST search (Altschul et al., 1990; Johnson et al., 2008) with subsequent filtering was performed on all NACs.

As a last step, the corresponding nucleotide sequences associated with unclear NACs were compared to the human reference genome with the **UCSC!** genome browser (Kent, 2002). Here, individual quality control based on different variant properties (position of mutation with respect to the localization of the exon/intron/splice site given by the primary transcript, mutation type, known transcripts, etc.) was performed, and peptides were rejected according to assessment.

All peptides with sequences of 8-15 AAs passing the procedures mentioned above were then assessed for in-vitro immunogenicity. This experimental testing was performed as described in section 2.3.2.

The principal work of this thesis consisted of developing scripts to annotate, analyze, correlate, and display mutational and peptide-specific data, to design prioritization methods based on BLAST and on results of in silico binding affinity prediction and apply them to the group of NACs. Finally, potential peptides were assessed for immunogenicity with interferon-based T-cell stimulation assays.

3.1.2 Characterization of the mutanome

The genomic and transcriptomic data set contains information on both, the genetic variants on DNA and those on RNA level. This data not only contributes to subsequent identification of potentially reactive neoantigens (NAs), but it also helps to develop a more accurate picture of the mutational landscape of the tumor.

The performed assessments included the evaluation of sequencing parameters, the DNA and RNA overlap, for comparison of repertoires, the genetic background of the identified variants for characterization and the analysis of variants shared between different patients and different metastasis of the same patient.

3.1.2.1 Assessment of sequencing parameters

As a first step, the sequencing parameters (see Section 2.2.1) were assessed. This helped to contextualize the obtained variant data within the identification pipeline and led to the evaluation of plausible filtering criteria (cp. Eq. 2.2-2.4).
For details on filtering criteria, see Methods Section 2.2.1.4. All variants not complying with one or more of these filtering criteria were termed "outliers". The remaining variants were termed "inliers".

Coverage For DNA variants, the mean value of total reads of all identified variants was roughly 140 (25^{th} to 75^{th} percentile ranged from 70 to 220 reads; Fig. 3.2). Considering the

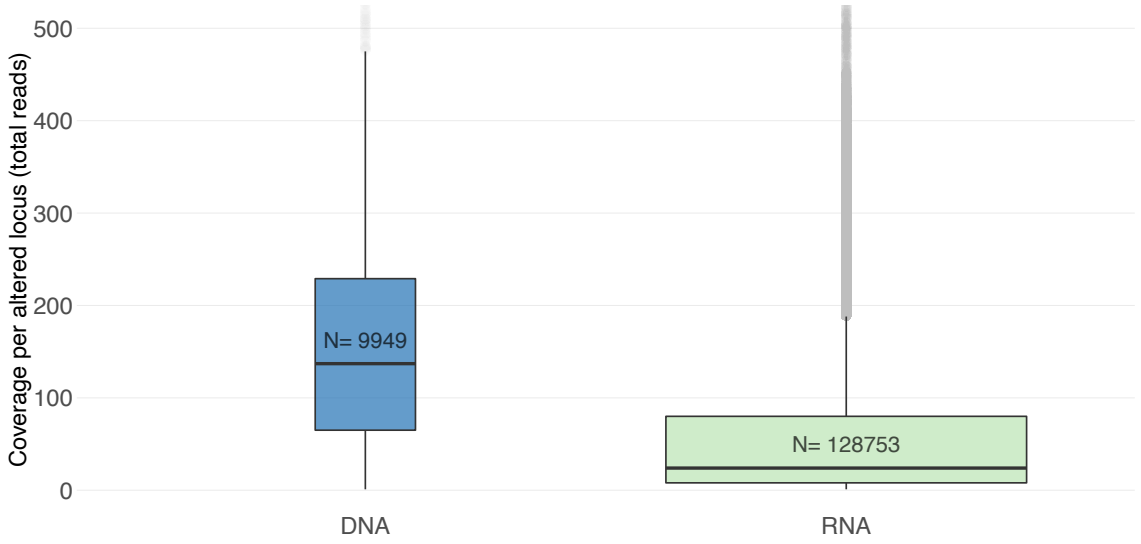


Figure 3.2: **Coverage per altered locus for both DNA and RNA variants.** The spread of the number of total reads (coverage) for all identified mutations is separately displayed for DNA and RNA level utilizing a boxplot (DNA: blue, RNA: green). A black horizontal line depicts the corresponding median of both distributions. The corresponding whiskers are displayed by two black vertical lines for each boxplot. Grey dots depict outliers.

alterations on RNA level, the mean value was found to be shifted towards lower values. A closer look revealed that even if the majority of identified RNA variants had fewer total reads (25^{th} to 75^{th} percentile ranged from 5 to 80 reads), distribution outliers were found at high coverage values. Per average, RNA variants exhibited lower sequencing depths. This impression could be confirmed by looking at the detailed histograms for coverage of DNA and RNA (Fig. 3.3). Here, most identified variants on DNA resembled a Gaussian distribution with a peak roughly at 180 reads. In comparison on RNA level, the identified alterations did not exhibit such a clear canonical distribution.

While for DNA, both the filtered ("Inliers", blue) and the unfiltered ("Outliers", yellow) variants exhibited a qualitatively similar distribution (Fig. 3.3), for RNA, two different peaks could be observed: The majority of inliers ranged between 5 and 100, outliers however, were either found to be below the threshold ($N_{total\ reads} = 5$, red dotted line) or between 100 and 300 reads.

Tumor VF In terms of tumor **VF!** (cp. Section 2.2.1.3), the RNA distribution was much wider and had a mean value of 0.25, whereas for DNA, more than 50% of all identified

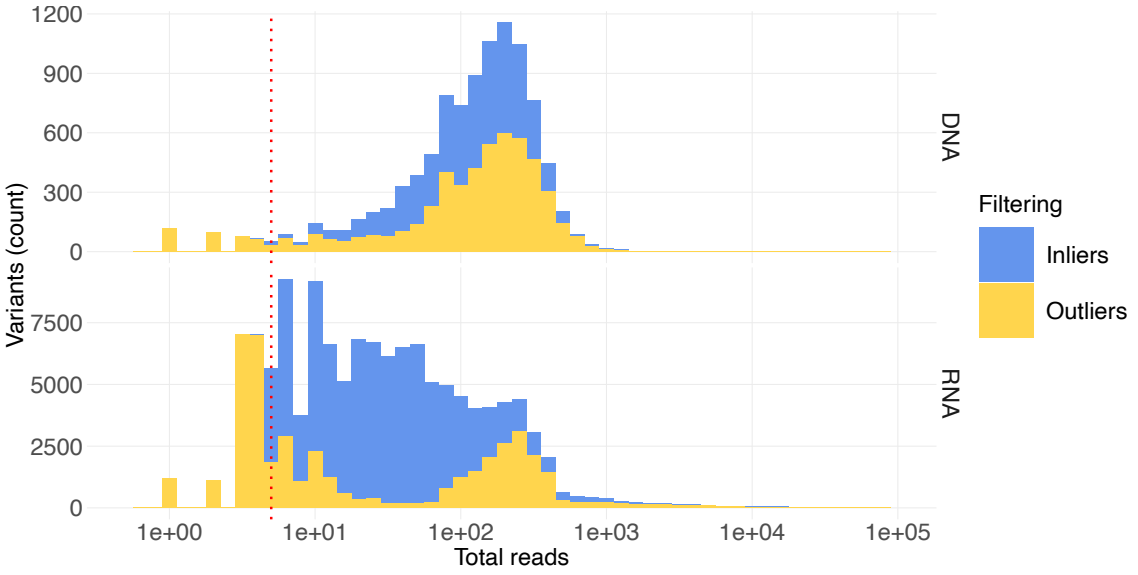


Figure 3.3: **Histogram of the distribution of the coverage per altered locus for Inliers and Outliers, for DNA and RNA level.** The distribution of the number of total reads per identified variant is displayed as a histogram for all variants meeting the filtering criteria (Inliers: blue) and for the remaining variants not meeting the criteria (Outliers: yellow). The upper panel depicts the distribution for DNA variants and the lower panel for RNA variants. The red dotted vertical line indicates the threshold for the filtering criteria.

variants had a tumor **VF!** below 0.25, with a mean value of approximately 0.08 (Fig. 3.4).

A substantial part of all variants, but especially of those detected on DNA level (Fig. 3.5, upper panel), yielded a tumor **VF!** below 5% (red dotted line) and were classified as "outliers" (yellow). Interestingly, the majority of variants on exome level with a tumor VF of 1 were rejected due to low coverage. In contrast, on RNA level, less than 50% of these high-VF alterations (and also a significant proportion of alterations at specific tumor VF values that correspond to certain fractions arising from low read numbers) were filtered out. For low and intermediate values of tumor VF, there was a broad distribution of variants passing the filtering criteria (Fig. 3.5), for DNA and RNA. Relatively fewer variants were identified with a tumor VF of > 50%.

3.1.2.2 Assessment of DNA and RNA overlap

After eliminating duplicates, the data set contained a total of approximately 139.000 variants thereof, 9.900 DNA variants (7.1%), and 128.000 RNA alterations (92.9%). Since RNA data was not available for all tumor samples, the analysis of cross-level variants (variants

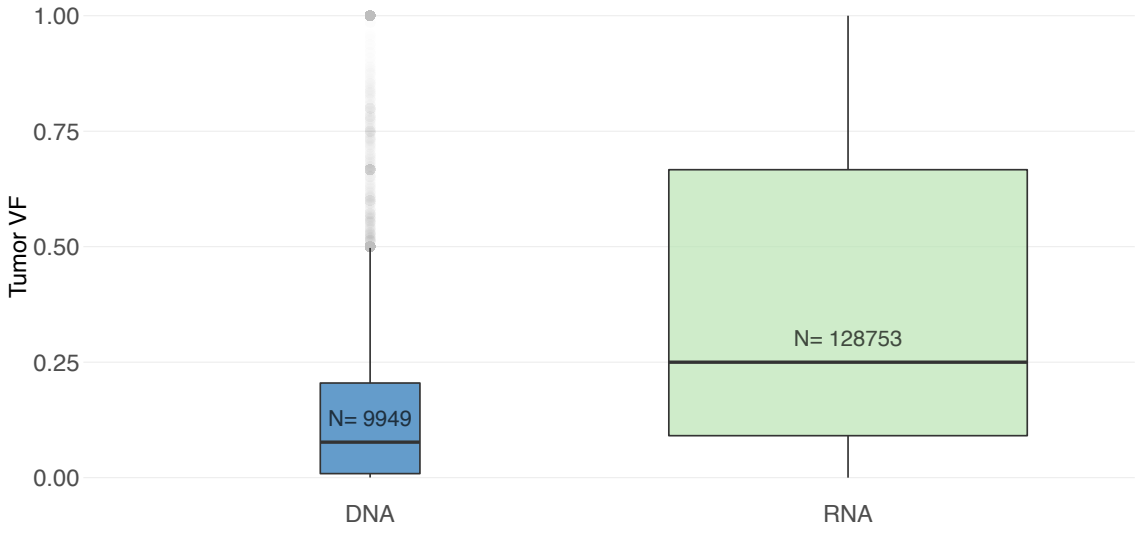


Figure 3.4: **Tumor Variant frequency for both DNA and RNA variants.** The two boxplots depict the distribution of the tumor VF of all identified variants on DNA and RNA level. Although the median, displayed by the horizontal black line, on transcriptome level is significantly higher than on exome level, the interquartile range on RNA level covers more than half of the whole range, indicating that the distribution here is much wider.

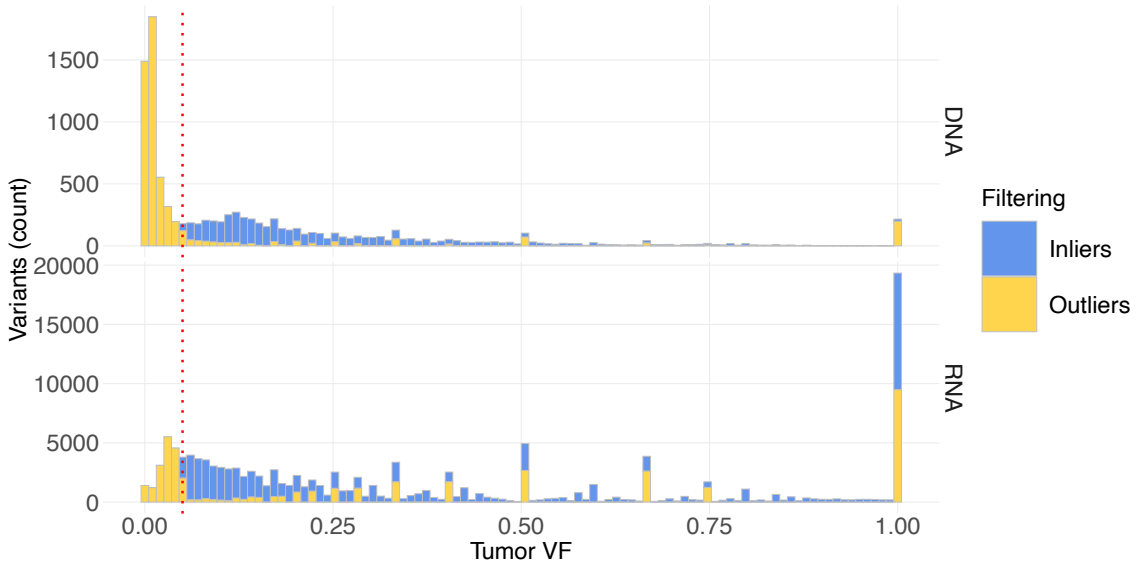


Figure 3.5: **Distribution of the tumor variant frequency depending on the filtering criteria for DNA and RNA variants.** The histogram depicts the distribution of the tumor VF for all identified variants with regard to the filtering criteria (blue: Inliers, yellow: Outliers) for DNA level variants (upper panel) and RNA level variants (lower panel). The red dotted line indicates the threshold for filtering criteria.

identified on RNA and DNA level) was based on a data subset containing only those samples with cross-level data availability. For this, the following tumor samples were excluded from the analysis: *11_T1, 16_T1, 20_T1, 34_T1, 31_T1, 14_T1, 25_T1, 25_T2*.

From the resulting 132.000 total variants, more than 5000 variants were identified in both data sets (Fig. 3.6), resulting in an overlap of 3.8%. By application of filtering criteria, as described in Section 2.2.1.4, 56% of all DNA variants and 35% of all RNA variants were classified as "outliers" (Fig. 3.7). This highlights that although having a much higher number of total reads, the exome data set had a much higher percentage of outliers, basically due to a vast fraction of variants with a very low tumor VF. This finding could also be confirmed by acknowledgment of the high peak in the histogram (Fig. 3.5, upper panel), which is located below the 5%-threshold (red dotted line) and represents all the outliers that were rejected due to low VF.

Next, the number of genetic variants was assessed for all patients with respect to their specific disease entity. For both levels of variants, DNA and RNA, the distribution did not show correlations between tumor entities and mutational burden (Fig. 3.8). It could be observed that the number of variants found for different metastasis and hence different tissue samples from the same tumor did not show divergent patterns but a major overlap.

The comparison of DNA and RNA reads yielded a substantial difference in the size of the corresponding subsets (cp. Fig. 3.6). The number of RNA variants was found to be approximately ten times higher than the number in the corresponding DNA data subset. When considering the applied filtering criteria (for details, see Section 2.2.1.4), the relevant group of exome variants was found to be even more diminished in comparison to the group of filtered RNA alterations.

To address the question of mutually shared DNA and RNA variants for different subgroups with similar entities, parametrized subsets of variants had to be compared in terms of congruence. Due to the heterogeneous cohort with 26 different entities among 32 patients, a further assignment into four disease groups ("Carcinoma", "Sarcoma", "Melanoma", "Other") was performed to categorize the results and to facilitate the deduction of possible correlations or dependencies.

As before, (cp. Fig 3.6) all identified variants were assigned according to their detec-

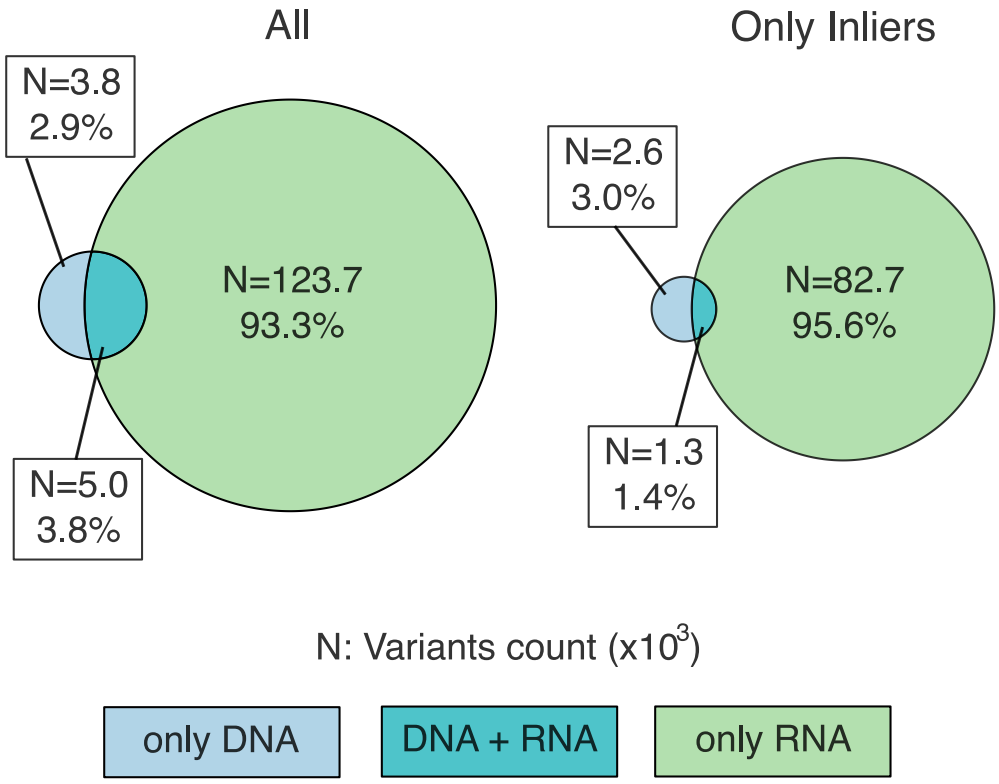


Figure 3.6: **Venn diagram showing the overlap of variants identified on DNA and RNA level for all variants and for Inliers only, respectively.** Each Venn diagram shows the three fractions of unique variants identified either on DNA (blue), on RNA (green), or on both levels (DNA and RNA, teal). All variants (left panel) were compared with those fulfilling the filtering criteria (right panel). Only tumor samples with available DNA and RNA data were considered for this analysis.

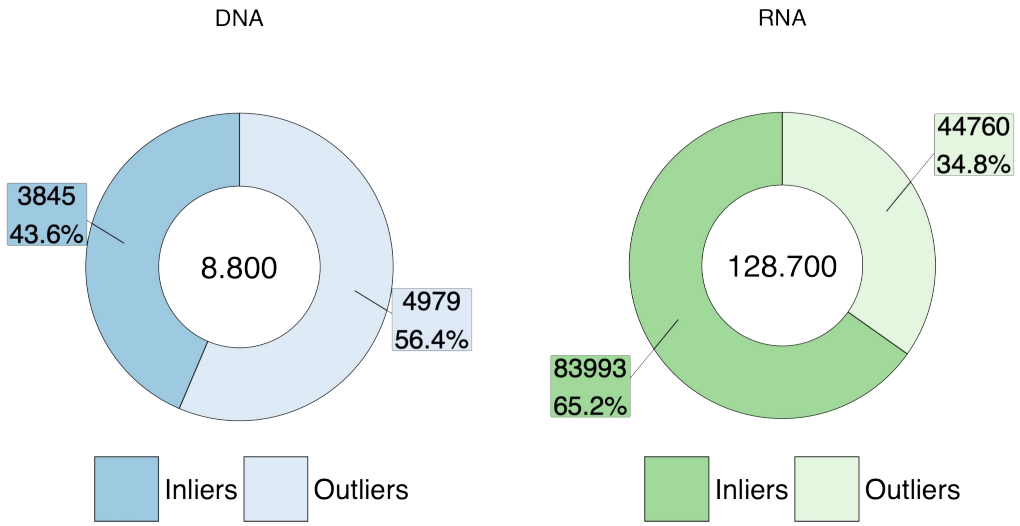


Figure 3.7: **Fraction of DNA and RNA variants classified as Inliers and Outliers, respectively.** The corresponding fractions of variants for both levels (DNA and RNA) that were assigned to the group of Inliers (darker color) and Outliers (lighter color) are displayed together with the corresponding absolute numbers of variants. Left panel: DNA, blue; Right panel: RNA, green.

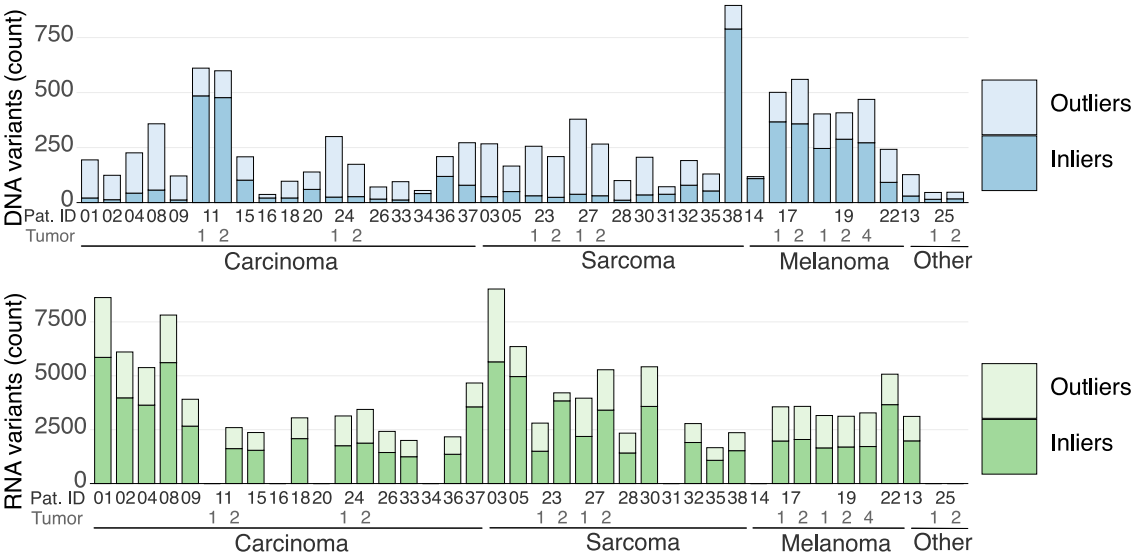


Figure 3.8: **Total number of genetic variants identified on DNA and RNA level.** For each sample and patient in the cohort, the total number of genetic variants identified on DNA level (upper plot) and RNA level (lower plot) is illustrated. By applying criteria on sequencing parameters, all found variants were clustered into a group of "Inliers" meeting the filtering criteria (darker blue/green) and a group of "outliers" not meeting the criteria (lighter blue/green). No RNA data was available for patients IN-11-T1, IN-14, IN-16, IN-20, IN-25, IN-31, IN-34.

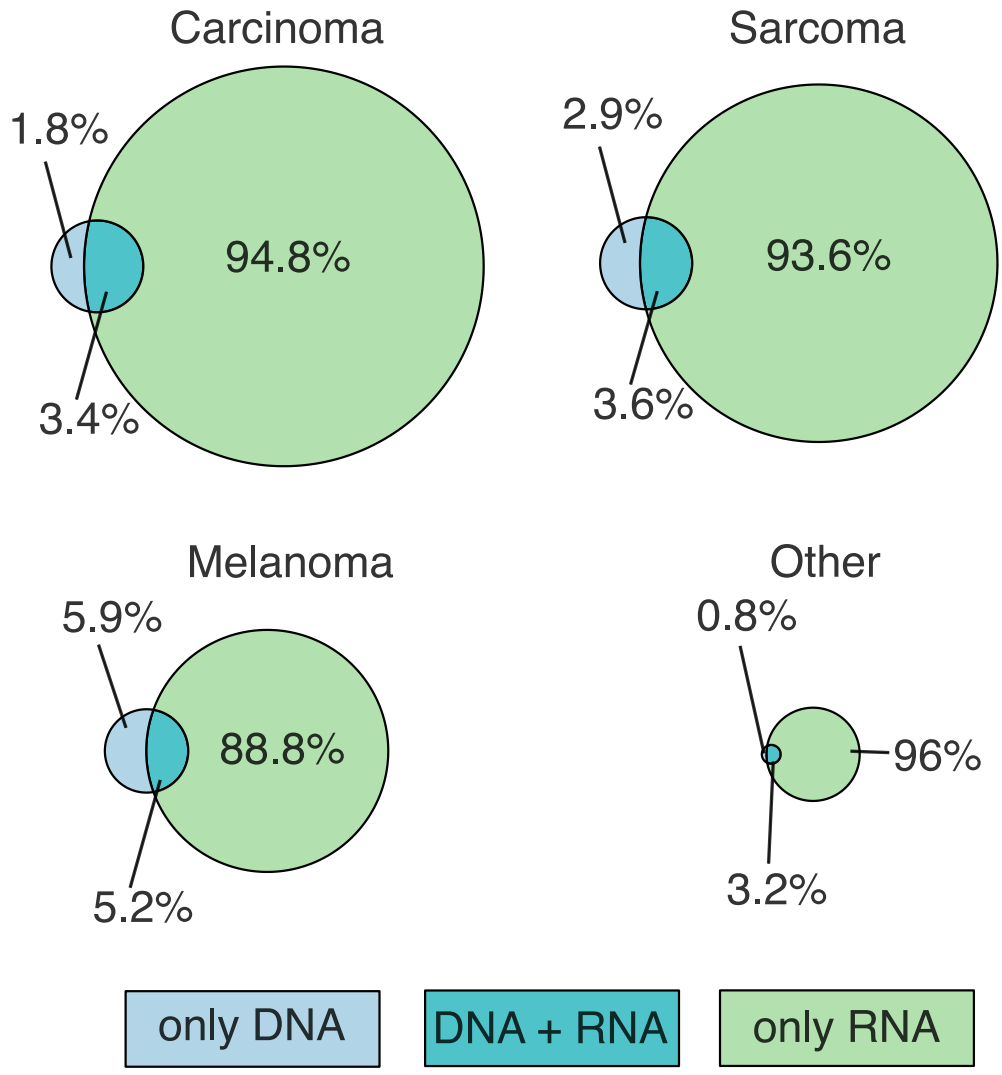


Figure 3.9: **Overlap of DNA and RNA variants for different subsets of entities considering all identified variants.** The different Venn diagrams show the overlap of shared DNA and RNA variants assigned to different subgroups associated with the tumor entity of the corresponding patient. The teal areas represent the overlapping fraction. The sizes of the circles hereby correspond to the absolute number of variants, whereas the percentages refer to the relative fraction within each specific subgroup. The biggest overlap (5.2 %) could be observed for variants identified in Melanoma patients.

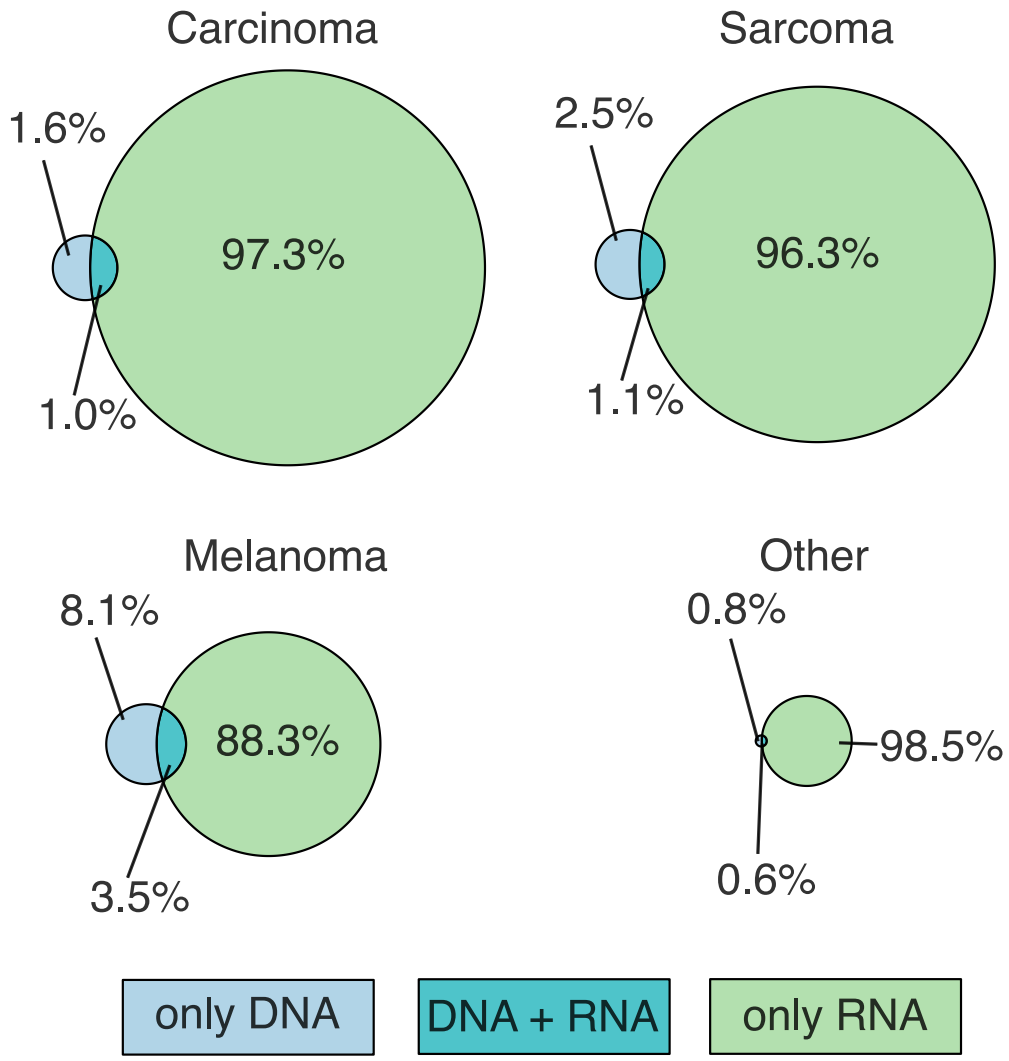


Figure 3.10: **Overlap of DNA and RNA variants for different subsets of entities considering only variants passing the filtering criteria.** The different Venn diagrams show the overlap of shared DNA and RNA variants assigned to different subgroups associated with the tumor entity of the corresponding patient, where only variants classified as "Inliers" were considered. The teal areas represent the overlapping fraction. The sizes of the circles hereby correspond to the absolute number of variants, whereas the percentages refer to the relative fraction within each specific subgroup.

tion level (DNA only, DNA + RNA and RNA only), but additionally clustered into each disease subgroup. It could be observed that for different subgroups, the relative overlap of variants varied significantly (Fig. 3.9). Whereas more than 5.9% of all variants identified in Melanoma samples (n=8) were only found on exome level, for samples of the carcinoma group (n=14), this value dropped to less than 1.8%, suggesting that variant coverage through RNA-Seq was even more effective in these cases. For the sarcoma group, intermediate levels of overlap were found.

Repeating the above analysis with the data set of filtered variants (see Section 2.2.1.4) indicated that the overlap of DNA and RNA variants is generally significantly smaller for inliers or that a significant part of outliers is shared between DNA and RNA (Fig. 3.10). This may have major implications for the emergence of immunogenic neoantigens and will be discussed in Chapter 4.

3.1.2.3 Classification and genetic assessment of variants

As seen before, the identification of genetic variants on the RNA level has led to an approximately ten times higher variant yield than on exome level. To address the question if post-transcriptional modifications, such as RNA editing, or if rather structural differences within the identification pipeline were responsible for this "variant-gap", detailed genetic analysis had to be made.

Genetic biotype The genetic biotype, that is, the classification of the genetic origin (the mutated gene) of all identified variants, was assessed with respect to the corresponding detection level. Here, only variants passing the filtering criteria (cp. Sec. 3.1.2.1) were taken into consideration (Inliers).
While more than 20% of the RNA variants resulted from regulatory RNAs (see Fig. 3.11), which was significantly higher than on DNA level (only 4% regulatory RNAs), the vast majority of variants for both regimes were detected on protein-coding regions. However, there was a notable difference for DNA variants (78% protein-coding) compared to RNA variants (54% protein-coding). Interestingly, processed transcripts, TECs, and sense intronics only played a minor role in terms of genetic biotype for DNA-identified variants (together <1%). For RNA variants, this fraction rose to roughly 7%.

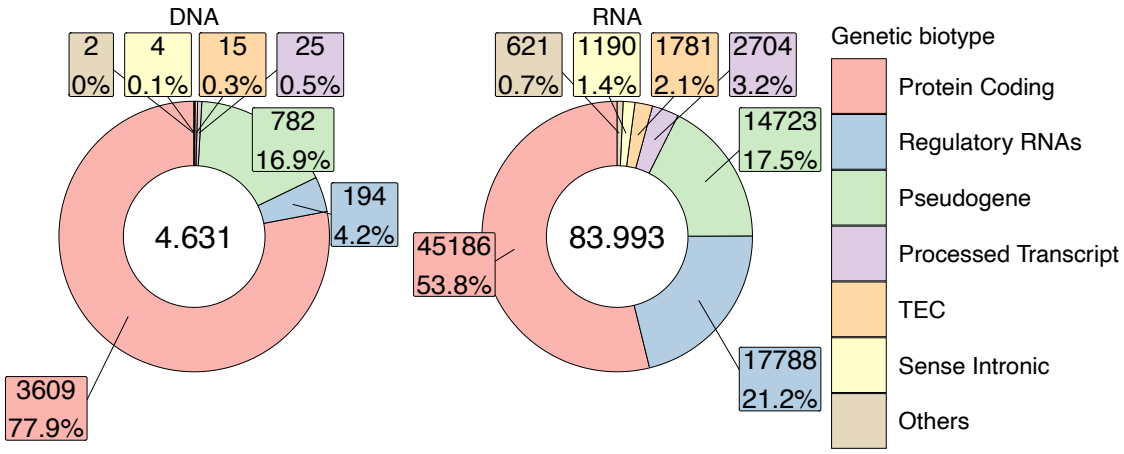


Figure 3.11: **Genetic Biotype of DNA and RNA variants.** The fraction of the most important genetic biotypes of all variants assigned to "Inliers" are displayed for DNA (left panel) and RNA (right panel) level.

Variant type To go more into detail, in a next step, the variant type itself was assessed by grouping all mutations into subgroups according to their genetic effect. It was mainly discriminated between non-coding and coding missense, splice site, intron, and other minor effects.

Again, assigning all variants to the DNA and RNA group revealed that for DNA, the coding missense variants were clearly dominating (64%), whereas, for RNA, most variants belonged to the group of non-coding missense variants (45%; Fig. 3.12). The fraction of coding missense variants on RNA level was only about 30%.

Remarkably, splice site and intron variants accounted for roughly 20% on RNA level but only 2% on DNA level. At the same time, frameshift events seemed more likely on DNA level (6%) than on RNA level (2%).

All variants assigned to other groups, taken together, were responsible for less than 3% for RNA variants and less than 6% for DNA variants.

Mutation type Further assessment, according to the associated mutation type of all variants, showed that roughly 89% of all DNA variants and 96% of all RNA variants resulted from substitution events (see Fig. 3.13).

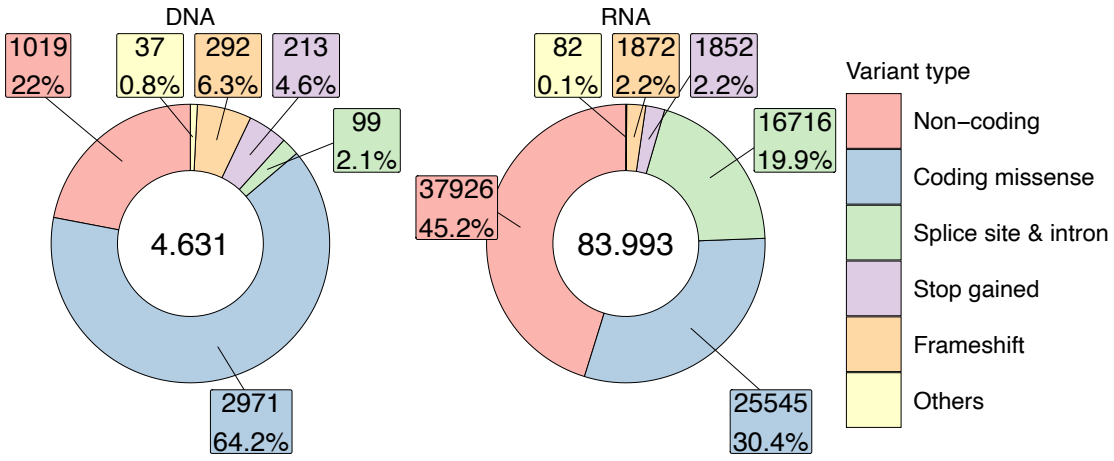


Figure 3.12: **Genetic Variant type of DNA and RNA variants.** The fraction of different variant types is illustrated for all variants meeting the filtering criteria for both, DNA and RNA level (left and right panel, respectively).

Only a substantially smaller fraction originated from deletions (DNA 7%; RNA 4%). Insertions and multi-substitutions appeared to be rare (DNA 4%) to extremely rare events (RNA <0.5%).

Gene mapping for RNA variants Mapping all identified variants to their associated genes was done to eventually highlight genetic conspicuousness for different tumors. First, all variants related to a certain gene were counted and weighted by the gene size, that is, the length of the gene in number of base pairs.

The 25 genes with the highest density of variants had a range of eight to 35 mutations per 100 bp (Fig. 3.14(a), blue and yellow bars). Considering only variants matching the filtering conditions (cp. Sec. 2.2.1.4; blue bars in Fig. 3.14(a)), there was no qualitative difference for the six most mutated genes. Either way, gene "RF00017" would still be the gene with the highest density of variants (followed by "AC018638.1 and "AC092718.4").

Interestingly, the fraction of variants being rejected by the filtering criteria varied dramatically between the different genes. Whereas variants from some genes (i.e., "SNRPGP15", "RPL24P2") seemed to be the result of nearly only inliers (passing the filtering criteria), some variants from other genes appeared to include a notable fraction of outliers (e.g., "AC099560.2").

For a more detailed data analysis, only variants matching the filtering criteria were in-

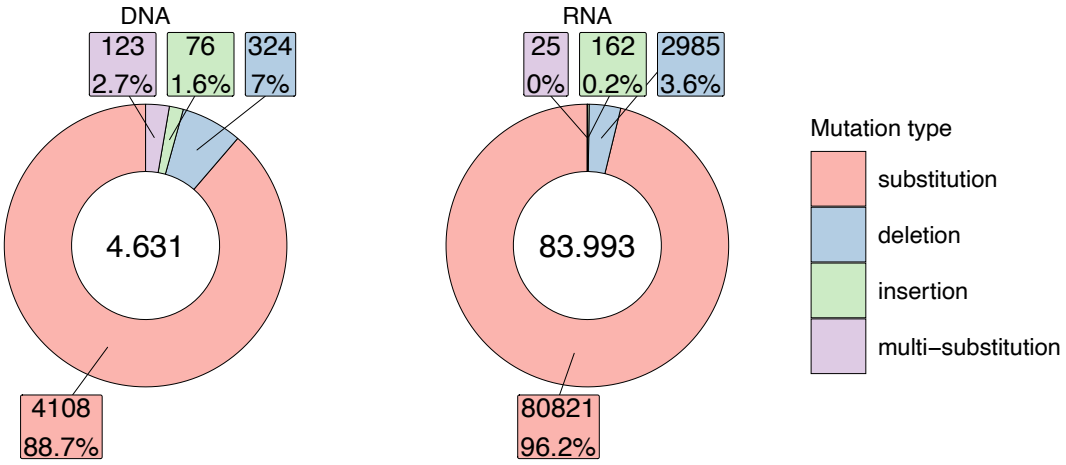


Figure 3.13: **Mutation type of DNA and RNA variants.** The doughnut plot depicts the different fractions of mutation types according to the level of identification (DNA variants: left panel; RNA variants: right panel). Substitution constituted the major fraction (88.7 % and 96.2 %, respectively). On RNA level almost no insertions or multi-substitutions were observed.

cluded.

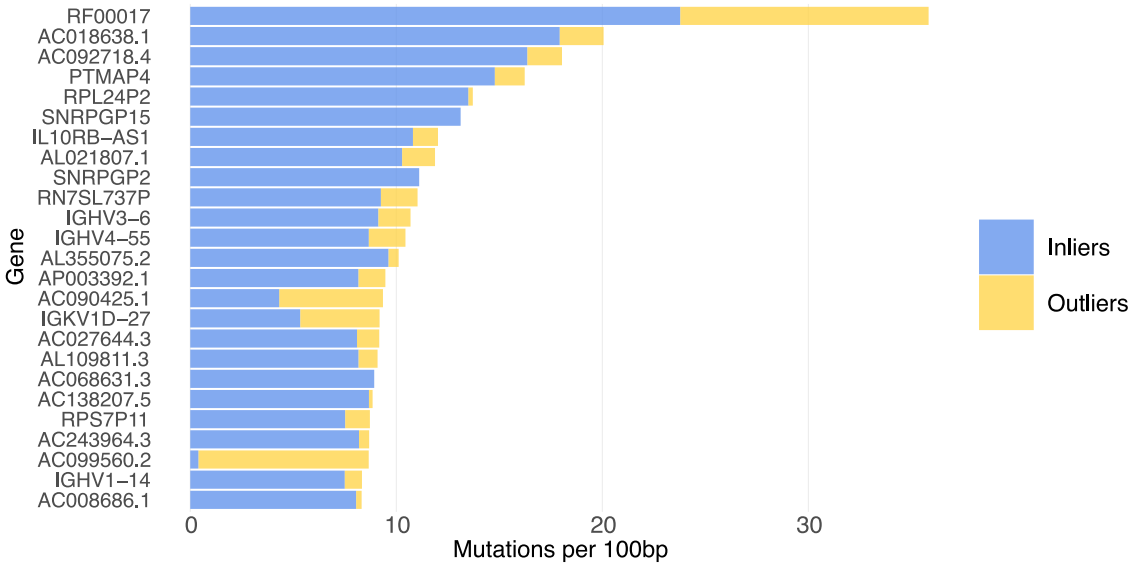
Specifying all variants by the tumor entity group of the corresponding sample, it could be seen that the individual fractions (according to the size of the sub-cohort) were comparable in size for most of the genes (see Fig. 3.14(b)). Nevertheless, some genes did not comply with this observation and exhibited a different behavior. First, variants in gene "RF00017" and "SNRPGP2" were exclusively found in Carcinoma-, Sarcoma- and Melanoma patients. Second, variants mapped to gene "IGHV3-6" were found to be significantly more abundant in tumors assigned to the Melanoma group than in other tumor types.

For an unbiased comparison of the most relevant (i.e., most mutated) genes between the different entity groups, the results were weighted by the occurrence in the particular group. The mutational load (i.e., the number of variants per 100bp per patient) in all three major groups was comparable with slightly increased levels in the case of the Melanoma patients (see Fig. 3.15).

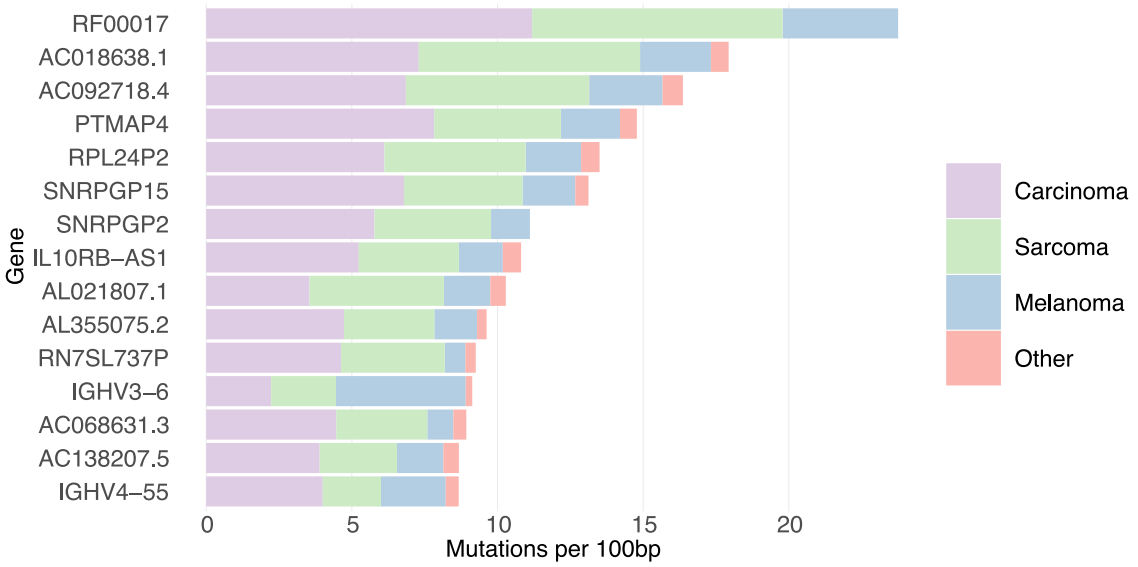
The mutational load for "Other" was found to be decreased roughly by a factor of two. Besides this, it could be shown that the three highest ranked genes ("RF00017", "AC018638.1", and "AC092718.4") were still within the first four genes with the highest mutational load for each of the three major groups (i.e., Carcinoma, Sarcoma, and Melanoma). Their val-

ues, however, varied significantly (for "AC018638.1") between 0.45 and 0.76 mutations per 100bp on average. For both, patients that did not belong to one of these three groups (denoted as "Other"), there were no variants identified within this gene.
According to the former observation of highly abundant variants in gene "IGHV3-6", this gene was ranked first for Melanoma patients with more than 1.1 mutations per 100bp per average. Interestingly, for the other entity groups, this gene was not ranked within the top ten genes and only reached a value of 0.11 to 0.22 mutations per 100bp on average. With one exception ("IL 10RB-AS1"), all genes that were found to be within the first 15 genes with the highest mutational load exhibited a gene length between 220 and 700 bp (see color distribution in Fig. 3.15).

Gene mapping for DNA variants Due to the significantly lower number of detected DNA variants compared to the transcriptome, ranking genes according to their variant density yielded a considerable number of very short genes, containing only one or two mutations. Hence, a threshold was introduced to avoid a certain selection bias due to a very low number of base pairs, which required genes to exhibit a minimum of three mutations to be considered in the analysis.
The bigger part of the genes had a roughly two orders of magnitude lower mutation density than the highest-ranked genes for RNA variants (see Fig. 3.16(a)). Strikingly, one gene ("DUX4L37") showed a mutation density that was increased by one order of magnitude compared to others. Curiously, it was found to carry this high number of variants only for the Carcinoma and the Sarcoma group (first bar in upper panels in Fig. 3.16(b)). The same held true for gene "HNRNPKP4", which again was found to only show mutations in patients from the Carcinoma and the Melanoma group.
In contrast and even more interesting, variants in gene "LINC00273" and "SALL1P1" could only be identified in patients from the Sarcoma and the Melanoma group (see Fig. 3.16(b)). For variants from the Melanoma group, gene "LINC00273" was even the one with the highest mutation density of all genes, with an average of 0.017 mutations per 100bp (Sarcoma: 0.014).
Higher-ranked genes were generally shorter (the four highest-ranked genes range between 700 bp and 1450 bp), whereas longer genes did not exhibit a correspondingly higher number of variants. In this regard, gene "FO538757" seemed peculiar since it carried more than 0.1 mutations per 100 bp with a gene length of almost 10.200 bp. For



(a)



(b)

Figure 3.14: **Genes with most RNA variants per locus.** The barplot depicts the mutational burden of the genes exhibiting the genes with the highest mutational variability per 100 bp. The total number of variants of a gene was weighted by its corresponding sequence length. **(a)**, The filtering condition's impact on each gene's mutational burden is illustrated. Here some genes have a strong bias towards unreliable variants, meaning that the majority of their identified variants did not meet the filtering criteria (i.e., "AC099560.2"). **(b)**, All Inliers were further classified according to their assigned entity subgroup, revealing some anomalies, i.e. for gene "IGHV3-6".

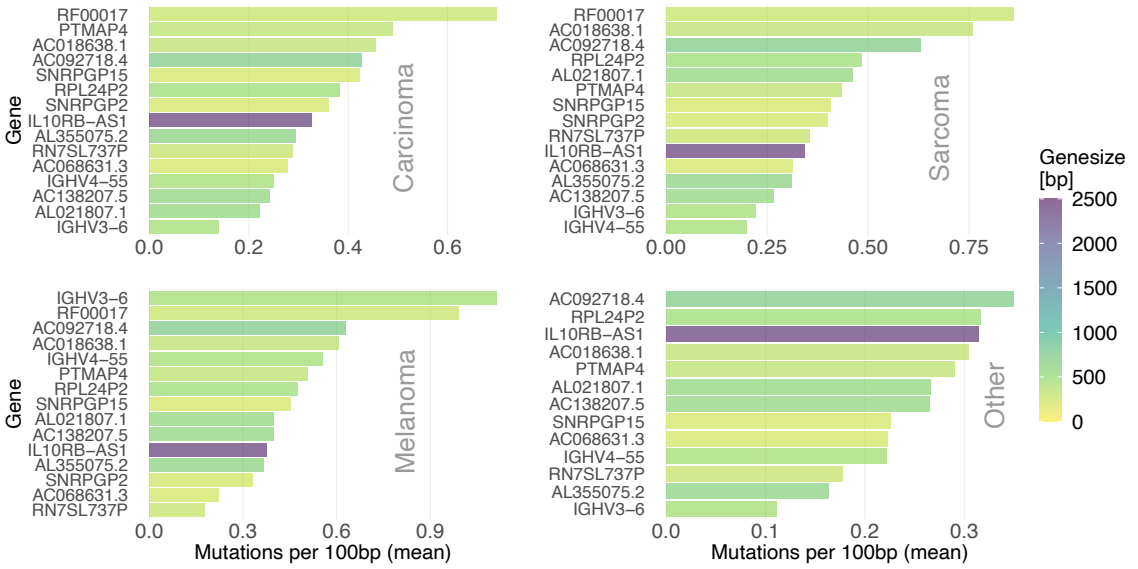


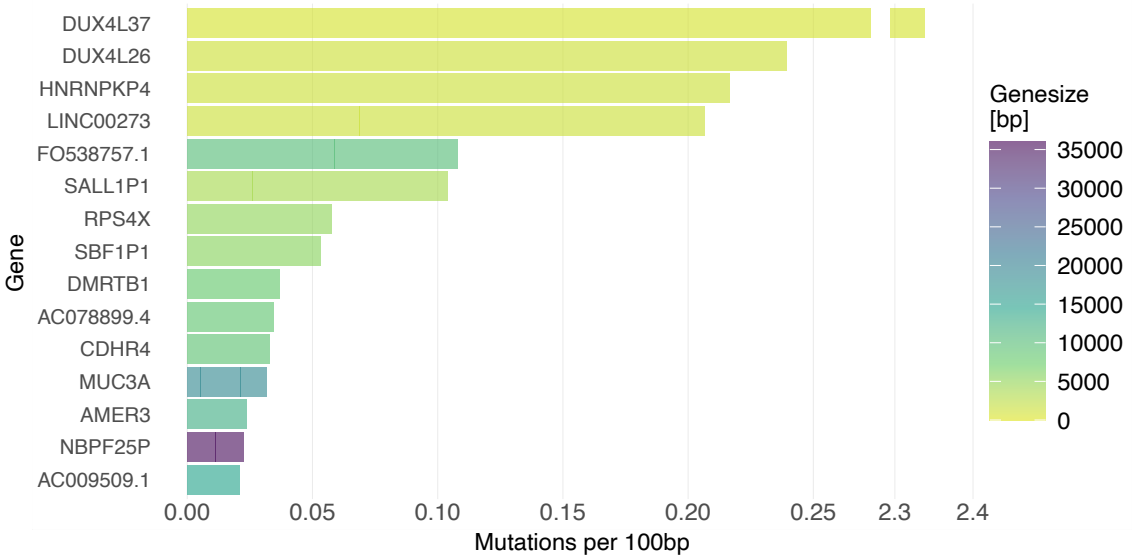
Figure 3.15: **Genes with the highest density of RNA variants for each entity group.** The 15 genes with the highest mutational burden are displayed for each of the four entity groups. For better comparability, the mutational burden is adjusted to represent the average value (mean) per patient and per 100 bp. The color code indicates the length of each specific gene. With one exception, most genes with high mutational burden had less than 1000bp.

patients assigned to group "Other", genes with the highest density of variants were, on average longer than 20.000 bp with no significant variant densities in short genes. Interestingly, gene "MUC3A", which was ranked for all groups besides the Carcinoma group, exhibited a mutation density that was roughly five times increased for patients of group "Other", compared to those of the Sarcoma or the Melanoma group (on average).

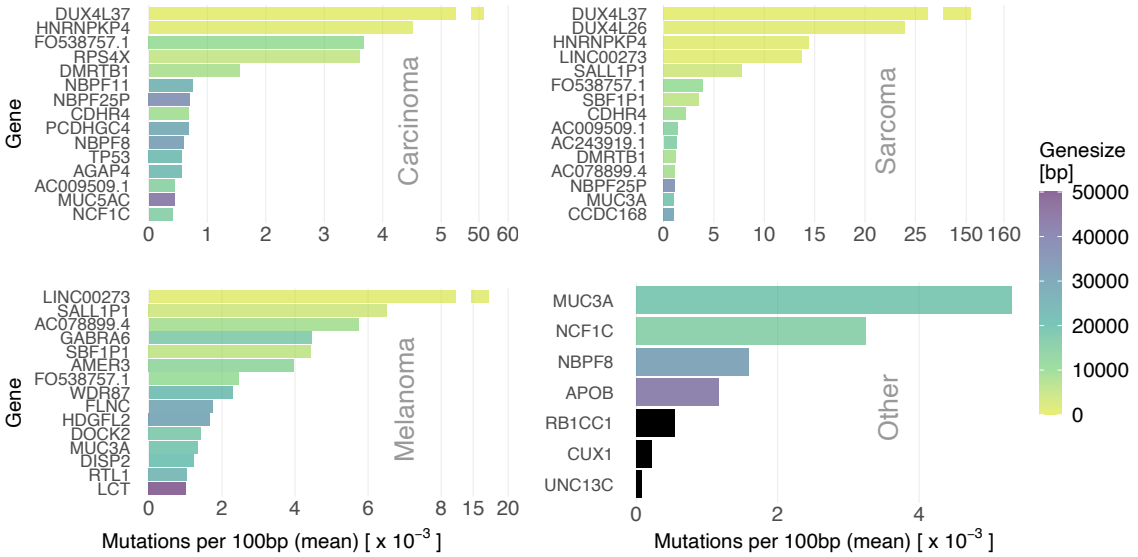
3.1.2.4 Variants shared between different samples

Another classification strategy, which can be used for pattern recognition of big data sets, such as the numerous variants identified in this pan-cancer cohort, is to rank subgroups of variants by the number of patients carrying the specific mutation. With this, it is possible to uncover entity-independent or crossover tumor-associated variants. These might be connected to neoantigens, representing highly attractive targets for immunotherapy. Besides, the assessment of variants for certain shared groups offers the possibility to discover variants that are associated, i.e., with specific tumor entities.

The assignment of all variants from both groups (DNA and RNA) to subsets according to

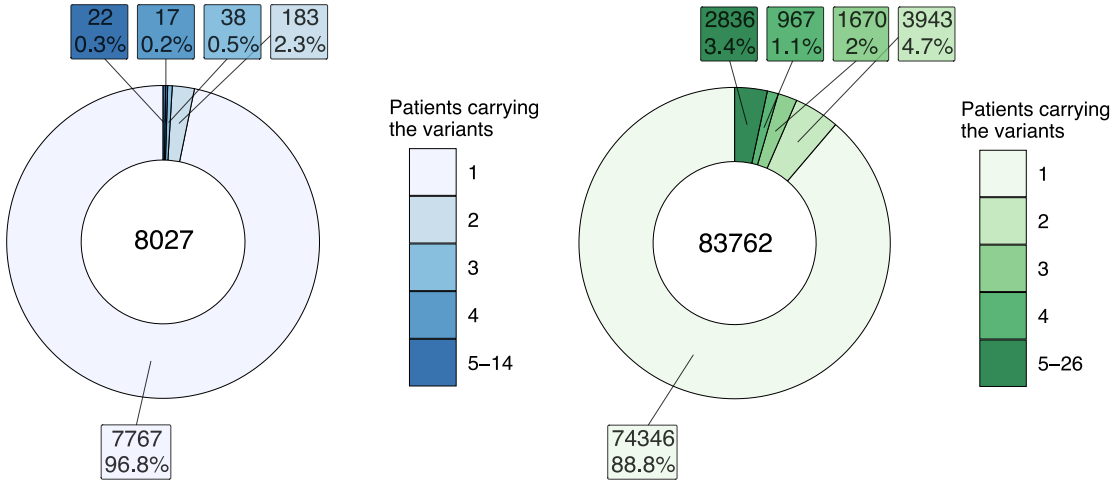


(a) Over all patients from the whole cohort



(b) Per entity group, weighted per patient

Figure 3.16: **Genes with the highest density of DNA variants. (a)**, The 15 genes with the highest mutational burden (density) are displayed together with the corresponding gene size (color code). Here all DNA variants ("Inliers", whole cohort) were considered (total value). **(b)**, For each of the four entity groups, the 15 genes with the highest mutational burden are displayed. For reasons of comparability, the mutational burden was adjusted such that it represents the average value (mean) per patient and per 100 bp. The color code indicates the length of each specific gene.

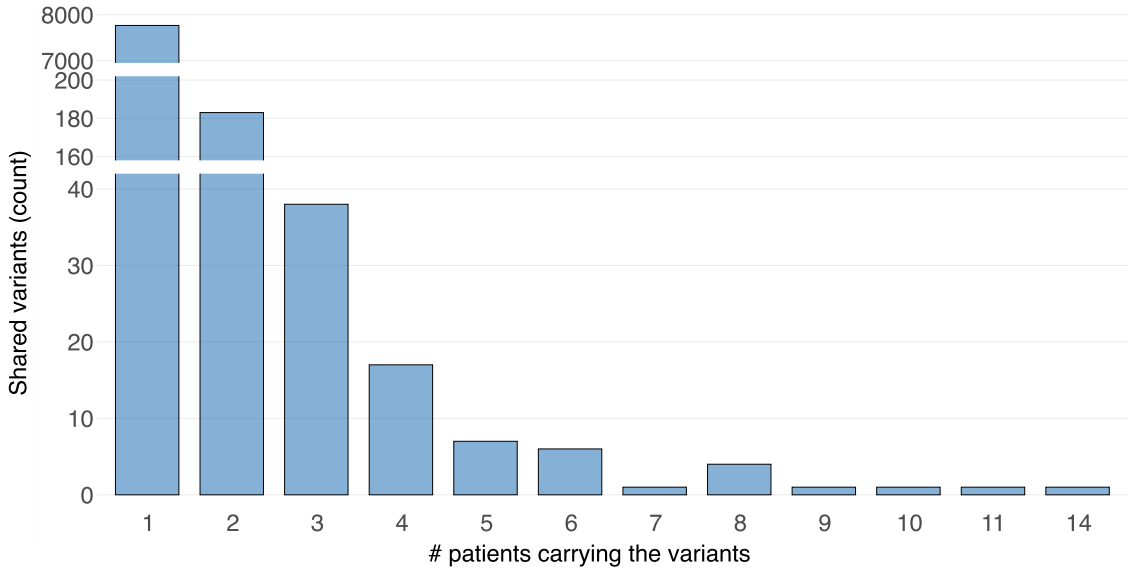


(a) DNA (b) RNA

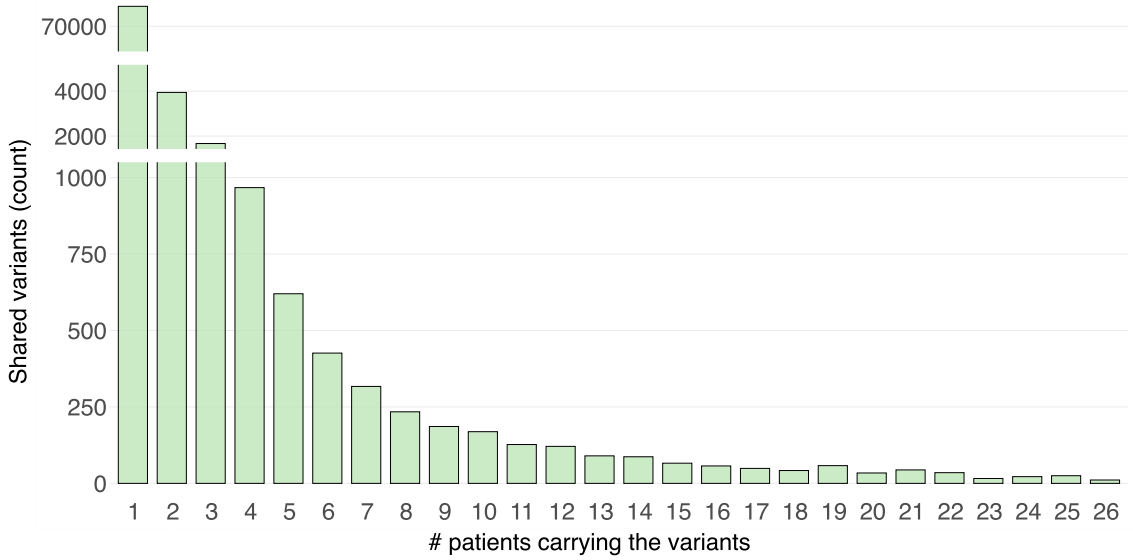
Figure 3.17: **Variants grouped by the number of carriers.** All fractions of variants according to the number of patients carrying this variant are displayed for DNA **(a)** and RNA **(b)** level. All variant subsets of more than four patients carrying this variant are grouped and displayed with one color (dark blue/green).

the number of patients that carry these variants revealed that the major part of all variants could solely be found in only one patient (see Fig. 3.17). Interestingly, this value differed markedly for variants detected on exome level (96.8%), and variants detected on RNA level (88.8%). With increasing set size, this means by going from sets of variants shared between fewer patients to those sets shared by more patients, the overall fraction of the corresponding variants decreased dramatically. Only 0.3% of all variants were shared by more than four patients in the DNA case.

The number of shared variants on RNA level dropped quickly with increasing set size (increasing numbers of overlapping patients). Still, it then remained relatively constant (see Fig. 3.18(b)), leading to a fraction of 3.4% of all variants that could be found in five or more patients (see Fig. 3.17(b)). Of note, RNA variants even shared between all 26 patients were detected. For DNA variants, only sporadic mutations were found in more than six patients, and no single variant could be found in more than 14 patients (cp. Fig. 3.18(a)).



(a) DNA



(b) RNA

Figure 3.18: **Number of variants grouped by subsets of patients carrying the variant.** All variants were grouped according to the number of patients carrying this specific variant, and the number of variants of each of these subsets is illustrated for all possible sizes of subsets. **(a)**, DNA variants. **(b)**, RNA variants.

3.1.2.5 Overlap of variants between multiple metastases of one patient

To investigate the phenomenon of tumor heterogeneity, it is inevitable to compare different metastases from the exact tumor origin and hence from the same patient. The whole cohort contained seven cases of tumor samples from the same primary tumor as well as different metastases. DNA and RNA data were available in six cases, so an extensive analysis of variants shared by different tumor manifestations could be performed.

Assignment of all unique variants to either one of the available, both two or (in the case of *patient 19*) all three metastasis, facilitated to display the corresponding overlap of variants identified on DNA (see Fig. 3.19), on RNA (see Fig. 3.20) and on both levels (see Fig. 3.21).

The fraction of the shared mutations on exome level (Fig. 3.19(b), purple area) was biggest for *patient 17* (Melanoma, 55% overlap) and relatively small for *patient 24* (Adrenocortical-CA, Fig. 3.19(e), purple area) and *patient 27* (Fibrosarcoma, Fig. 3.19(f), purple area), that both showed an overlap of approx. 12 %. As before, in the case of *patient 17*, for *patient 19* (also Melanoma), again there was a rather big mutational overlap of all three tumor manifestations (44%, see brown ellipse in Fig. 3.19(c)) with some minor intersection regions of variants shared by only two metastasis comprising between 2% and 4%.

Regarding the overlap of variants that were identified on RNA level, a similar picture could be observed: Again, the most remarkable overlap was found to be present in the Melanoma patient (*patient 17*, 42%; see Fig. 3.20(b)). This time, the overlap with the smallest fraction of shared variants on RNA level was found for *patient 23* (Rhabdomyosarcoma, 13% overlap). For *patient 19*, the triple-overlap decreased to roughly 28% with slightly increased double-overlap regions (varying between 4% and 7%; Fig. 3.20(c)). Since no RNA Data was available for tumor T2 of *patient 11*, no statement could be made here.

In addition, the overlap of variants found on both detection levels (DNA and RNA) was investigated. Due to a significantly reduced number of mutations meeting the demands, the evidence here was limited. Nevertheless, displaying the different shared fractions of variants yielded a comparable pattern. Both Melanoma patients (*patient 17* and *patient 19*) were found to hold the most shared variants between different metastasis (40% and

30%, respectively; see Figs. 3.21(b) and 3.21(c)). *Patient 27* and *patient 24* were found to have less than ten percent of variants shared between both metastases (7% for *patient 27* and 8% for *patient 24*, respectively; cp. Figs. 3.21(f) and 3.21(e)). Considering the symmetry (i.e., the balance) of the displayed shared fractions, it might seem peculiar that for *patient 19* (Melanoma), each overlapping, partly overlapping, and non-overlapping fractions were rather equal in size, meaning that shared variants were distributed evenly. The resulting picture is a well-balanced distribution with comparable intersections (Fig. 3.20(c)). It should be noted that in the case of *patient 11* both mutation sharing tumors had different tumor entities and were not metastasis of the same primary.

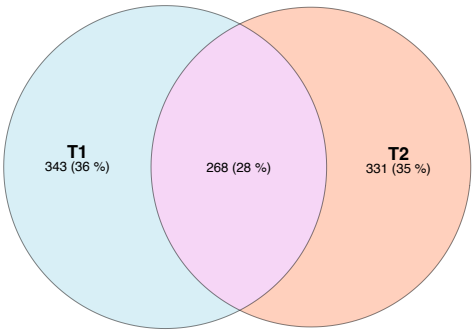
3.1.3 Assessment of selection criteria for peptide candidates

3.1.3.1 Prediction of peptide-MHC class I binding affinities

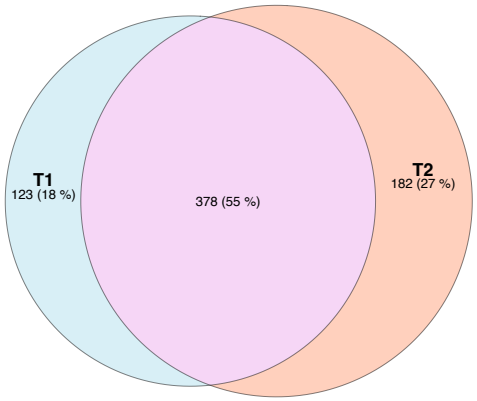
To identify altered peptide candidates, translated fragments from obtained sequencing data were used to predict their potential binding to defined HLA alleles. All binding affinity prediction was assessed as described in Section 2.2.3. The results for MHCflurry-based predictions (percentile rank) are displayed in Fig. 3.22, and the results for netMHC-based predictions (percentile rank) are shown in Fig. 3.23.

From these results, each NAC was assigned a best binding HLA allele with a corresponding best binding affinity (percentile rank and nano-molar affinity). For reasons of simplicity in the following only results for percentile rank will be illustrated. For the predictions, realized with *MHCflurry*, 53 out of 94 assessed NACs (see Appendix B, 2.1), had a predicted best percentile rank below 2% (weak and strong binders; cp. red dashed line in the lower panel of Fig. 3.24). Of these, 31 NACs were classified as strong binders with a percentile rank <0.5%.

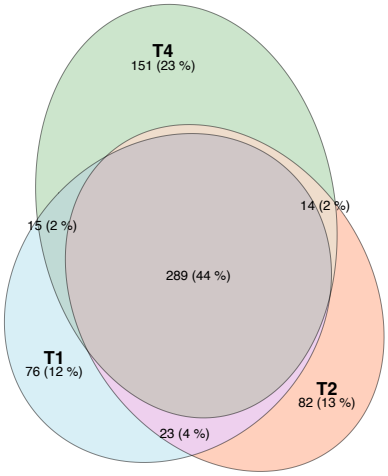
For the predictions done with *netMHC*, only 31 out of 94 assessed NACs had a best percentile rank value below 2% (weak and strong binders; cp. red dashed line in the upper panel of Fig. 3.24). Out of these, 17 NACs were classified as strong binders.



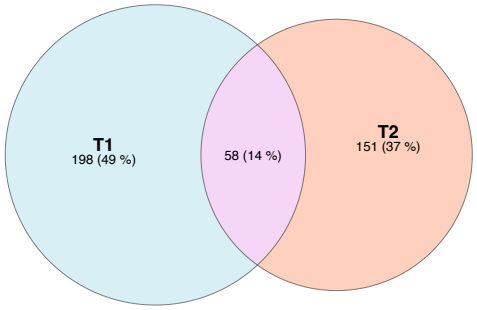
(a) Patient 11



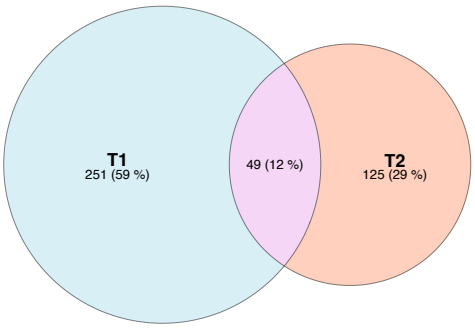
(b) Patient 17



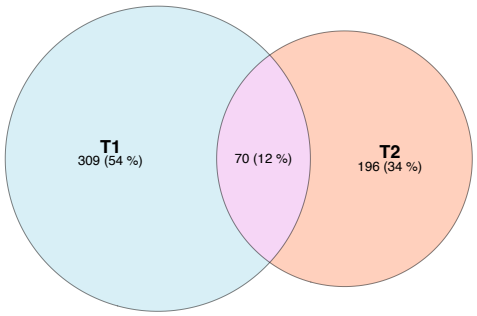
(c) Patient 19



(d) Patient 23

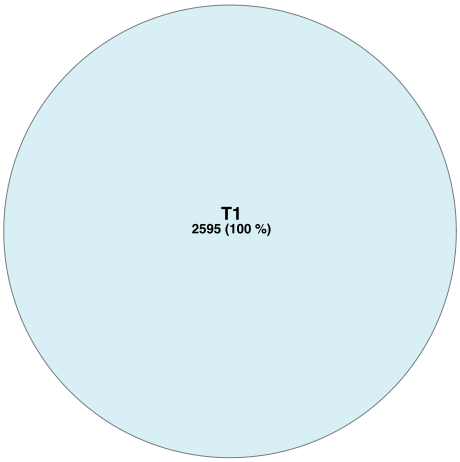


(e) Patient 24

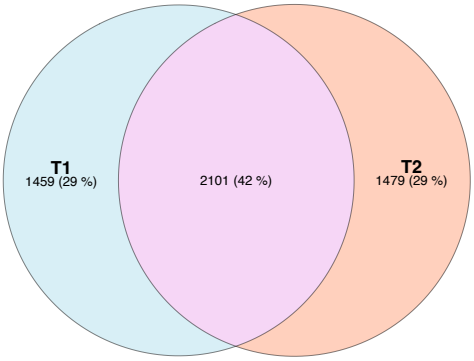


(f) Patient 27

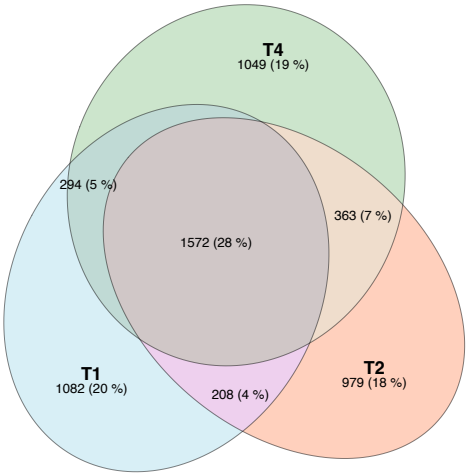
Figure 3.19: **DNA variants shared by multiple metastases.** The numbers and the percentages in the colored areas refer to the number of unique mutations and their fractions in the corresponding metastasis, respectively.



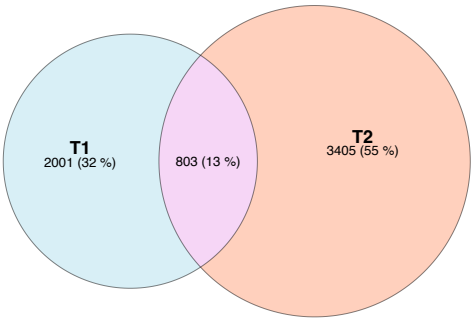
(a) Patient 11



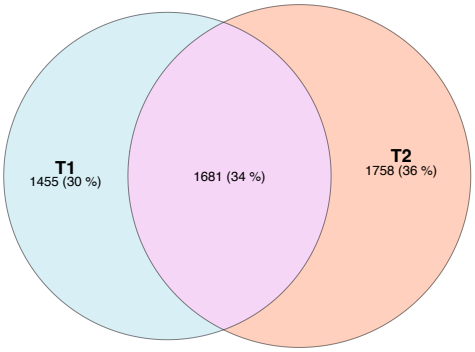
(b) Patient 17



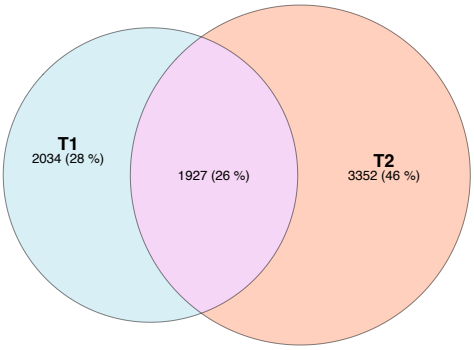
(c) Patient 19



(d) Patient 23

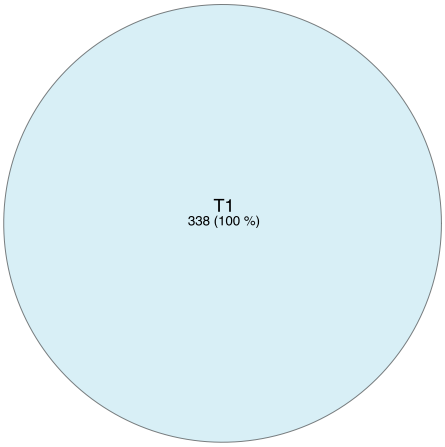


(e) Patient 24

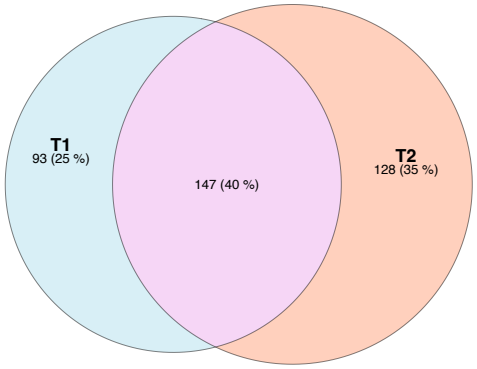


(f) Patient 27

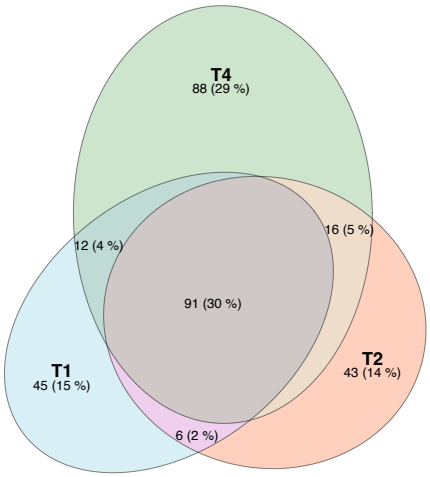
Figure 3.20: **RNA variants shared by multiple metastases.** The numbers and the percentages in the colored areas refer to the number of unique mutations and their fractions in the corresponding metastasis, respectively. **(a)**, For *patient 11* no RNA data was available for tumor T2.



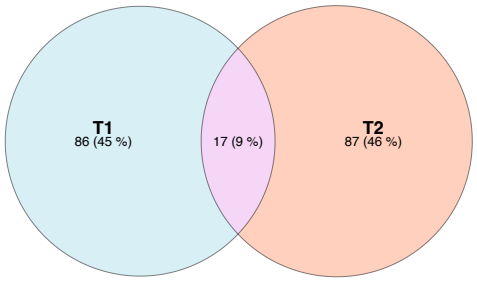
(a) Patient 11



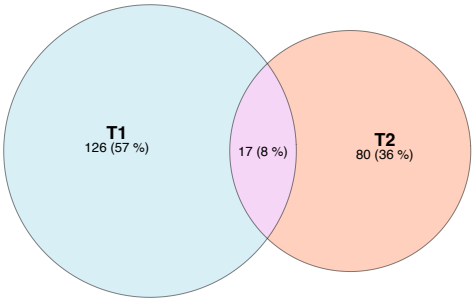
(b) Patient 17



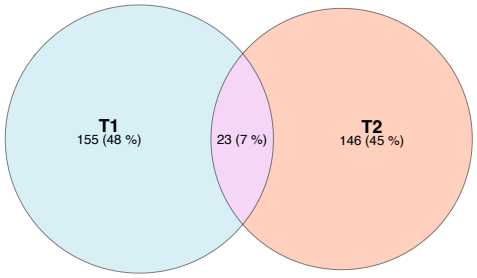
(c) Patient 19



(d) Patient 23



(e) Patient 24



(f) Patient 27

Figure 3.21: **Variants found on both, DNA and RNA level, shared by multiple metastases.** The numbers and the percentages in the colored areas refer to the number of unique mutations and their fractions in the corresponding metastasis, respectively. **(a)**, For *patient 11* no RNA data was available for tumor T2.

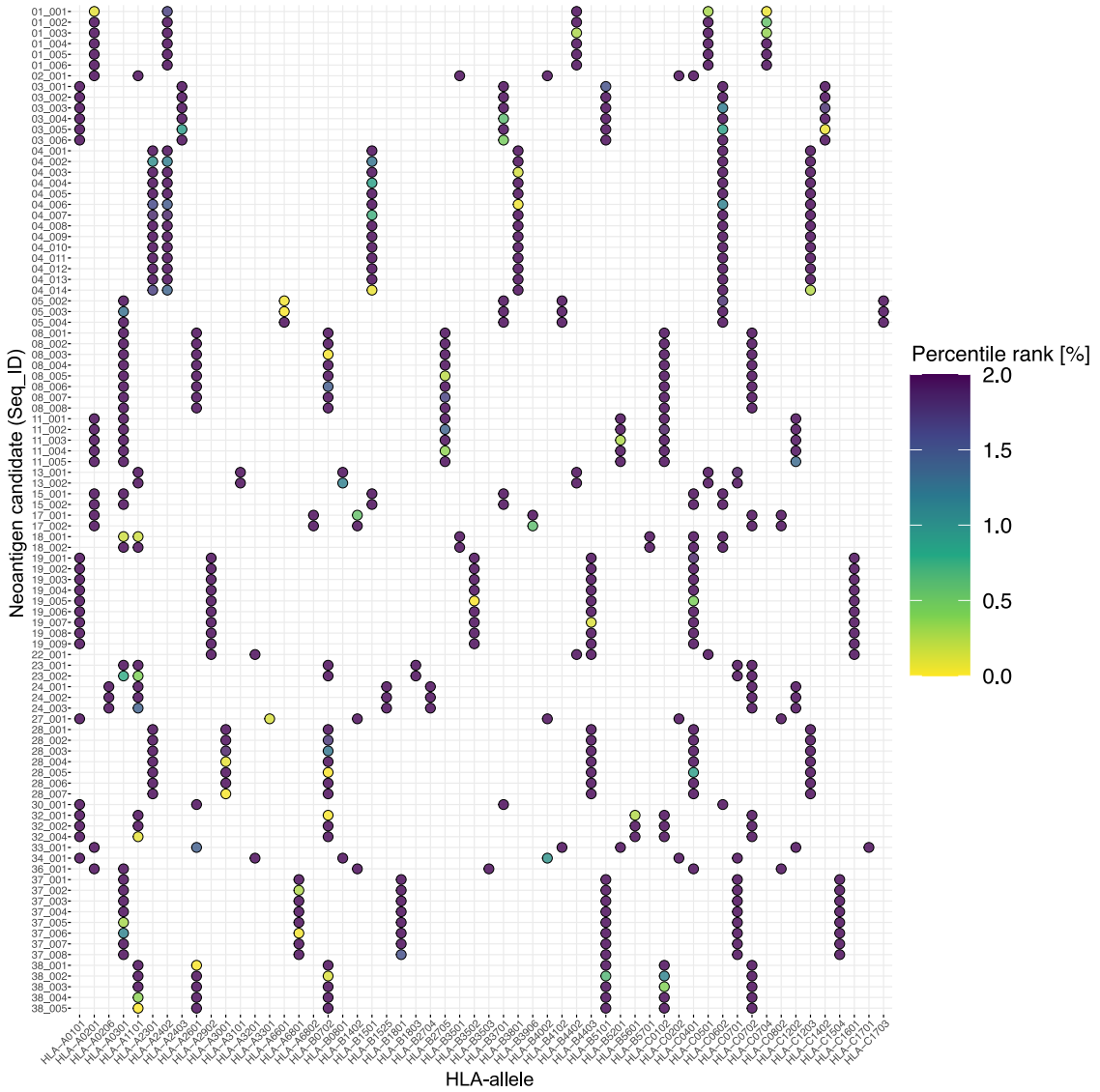


Figure 3.22: **Prediction of MHC class I binding affinities with MHCflurry.** Neoantigen candidates are displayed vs. HLA-I alleles. Every dot represents the predicted binding affinity for a NAC-HLA pairing. The color of the dot indicates the percentile rank. All values greater than 2% ("no binders"; cp. table 2.15) are colored equally.

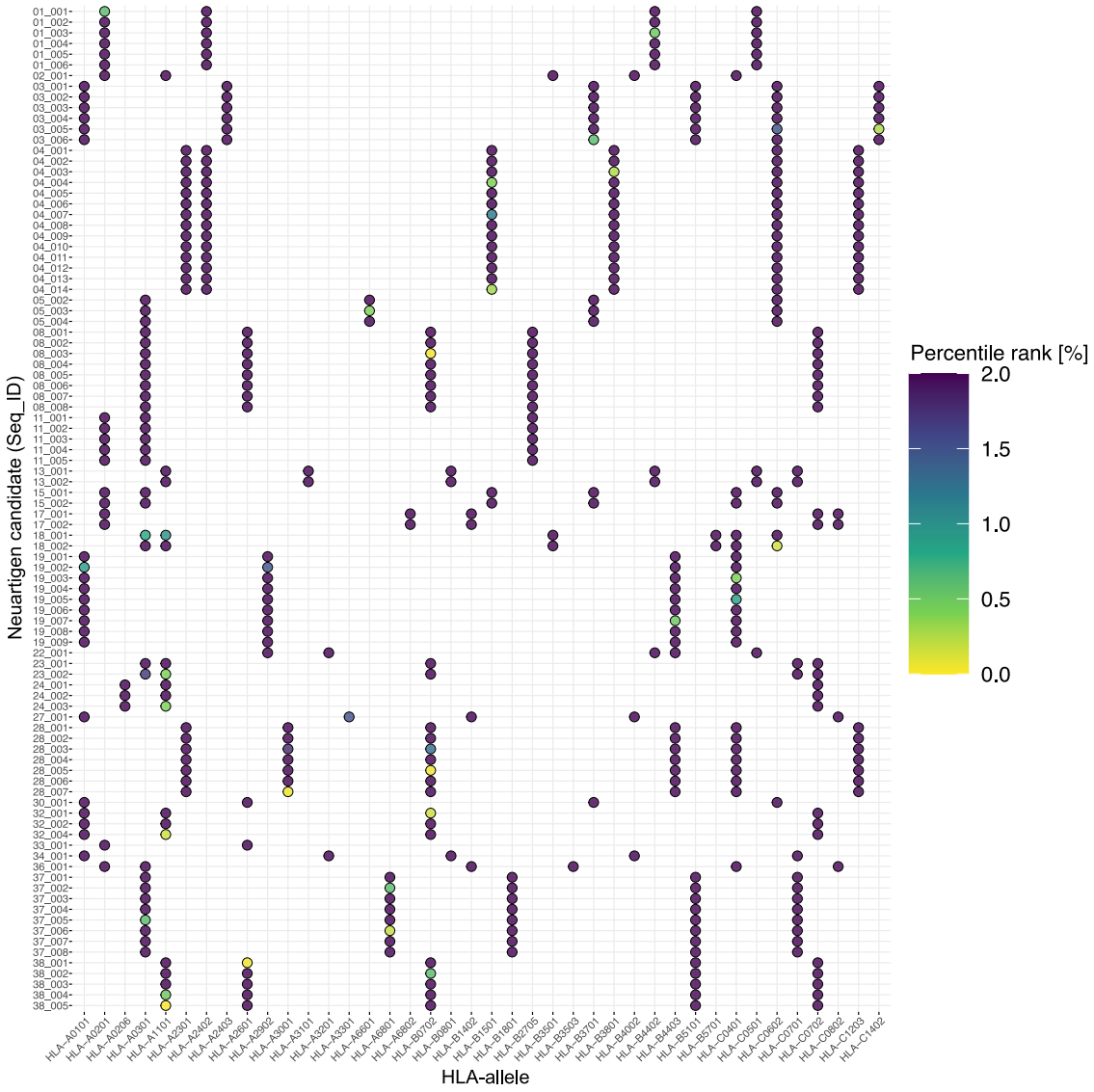


Figure 3.23: **Prediction of MHC class I binding affinities with netMHC.** Neoantigen candidates are displayed vs. HLA-I alleles. Every dot represents the predicted binding affinity for a NAC-HLA pairing. The color of the dot indicates the percentile rank. All values greater than 2% ("no binders"; cp. table 2.15) are colored equally.

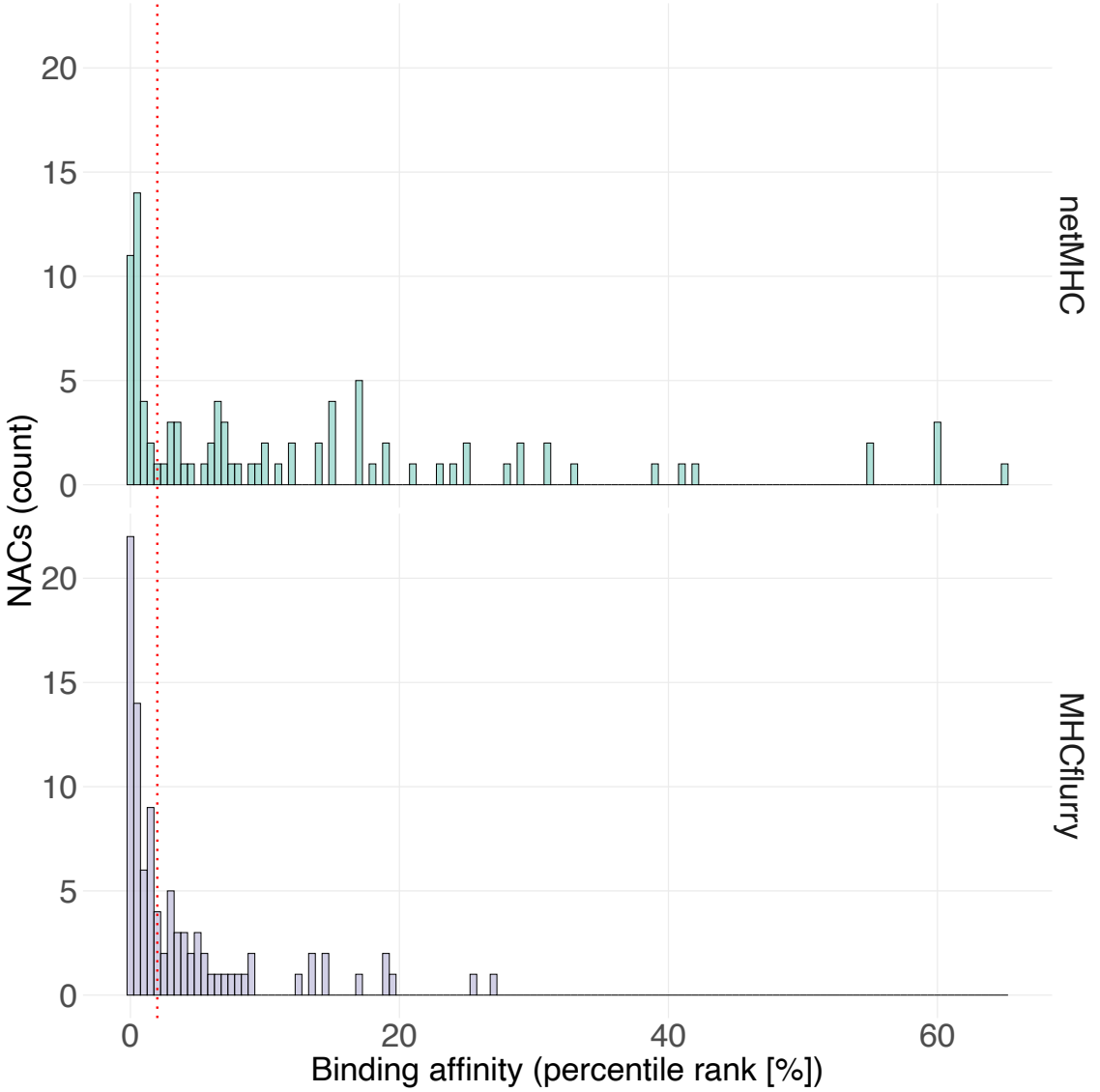


Figure 3.24: **Distribution of binding predictions (percentile rank).** The distribution of the percentile rank for all 94 NACs is displayed for netMHC (upper panel) and MHCflurry (lower panel). The red dashed line represents the threshold for weak binders (2%). Cp. table 2.15.

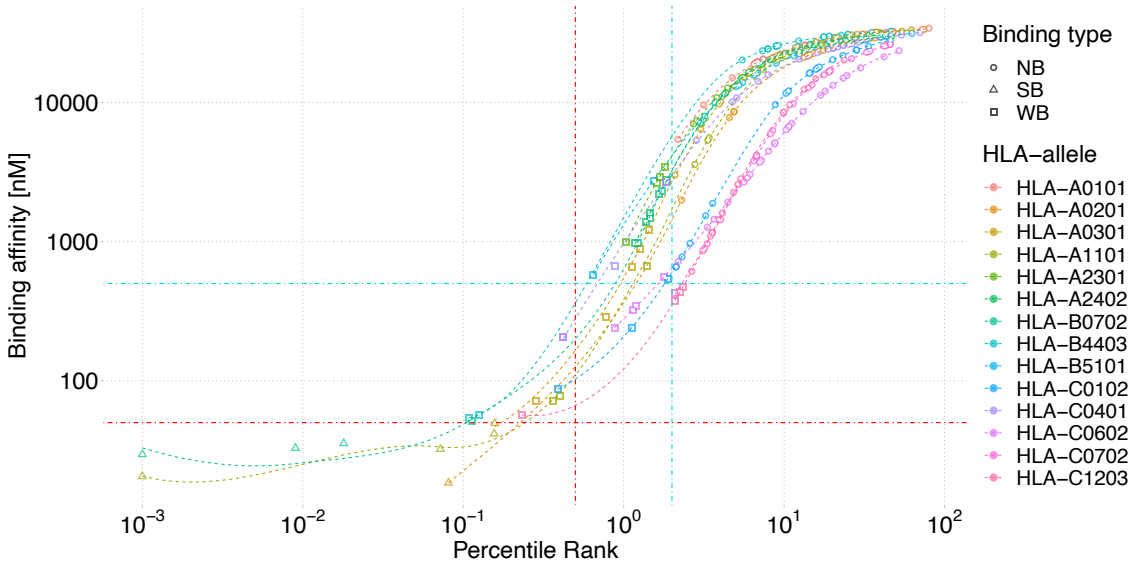


Figure 3.25: **Comparison of selection methods for MHCflurry.** The binding affinity of each NAC-HLA pairing is plotted versus its percentile rank. The resulting dots can be fitted with a sigmoidal regression curve revealing the nature of the dependence of both selection methods for the same HLA allele. The vertical and horizontal dashed lines correspond to the thresholds for weak (green) and strong (red) binders, respectively.

3.1.3.2 Comparison of rank vs. dissociation constant weighted binding affinity prediction

For each pairing of NAC-HLA allele, an in silico prediction with two different output methods was performed, yielding a rank and a nano-molar affinity. Comparing both selection methods by plotting the binding affinity versus the percentile rank for each composition revealed the allele dependence of their relation (see Fig.3.25). Every value was classified by the estimated binding type (see Tbl. 2.15; circle: No binder, square: Weak binder, triangle: Strong binder). All binders of the same HLA-allele could be fitted by a sigmoidal curve (dashed colored lines) specific for this predictor within the model. Each curve exhibited an allele-specific offset (horizontal shift) that led to a distribution width of roughly 0.6 orders of magnitude.

In the high-affinity regime, the regression curve deviates from the data points leading to higher fitting errors. The corresponding thresholds for weak and strong binders for both methods are displayed by horizontal (*Kd-method*) and vertical (*rank-method*) dot-dashed lines, respectively.

Next, the used prediction algorithms were compared. Therefore the results of the predictions with netMHC and MHCflurry were plotted in terms of a percentile rank and absolute binding affinity (Fig. 3.26). Each dot represents a NAC-HLA allele match, resulting in 580 possible combinations for the whole cohort.
Since netMHC did not provide modeling data for all identified HLA-alleles at the time of writing, predictions could only be determined for 478 of these pairings.

The Pearson correlation coefficient, used to describe the correlation between the percentile rank of netMHC and MHCflurry, yielded a value of $R = 0.81, p < 2.2 * 10^{-16}$, indicating a strong positive association. To visualize this correlation, a linear regression model, illustrated by the black curve with the corresponding 95% confidence interval, is displayed in Figs. 3.26(a) and 3.26(b), respectively.

By splitting up the regression line fitting according to the binding type classification (see Fig 3.26(a), cp. Tbl. 2.15, grey: No binder, green: Weak binder, red: Strong binder), some remarkable differences could be revealed. Whereas correlations for both prediction algorithms seemed to have similar slopes in the strong binding regime and for no binders, a negative slope was observed for those assigned to the group of weak binders. Here the predictions of the percentile rank seemed to differ significantly from the results of other binding types, indicating that netMHC estimates tended to higher values (i.e., bigger distances between data points and the red dashed line representing $x = y$).

A slightly diminished Pearson correlation of $R = 0.74, p < 2.2 * 10^{-16}$ could be determined for the correlation between both results of the nano-molar affinity (depicted by the dissociation constant Kd), which still indicated a strong positive association between both methods (see Fig. 3.26(b)).

By coloring the dots according to the binding type classification (realized according to the results of the percentile rank method) and, at the same time, plotting threshold lines for weak binders and strong binders (realized according to the results of the nano-molar affinity method; green, respectively red dash-dotted horizontal and vertical lines), major differences of both approaches could be uncovered. An apparent clustering of dots in the upper part of Fig. 3.26(b), seemed to deviate significantly from the expected regression line in the regime of intermediate affinities. In contrast, for very high affinities, both results appeared to be in good agreement (red dots in the lower left part of the figure).

Furthermore, a significant part of these aberrant dots was classified as strong binding pairings (percentile rank method), further emphasizing the big discrepancy.

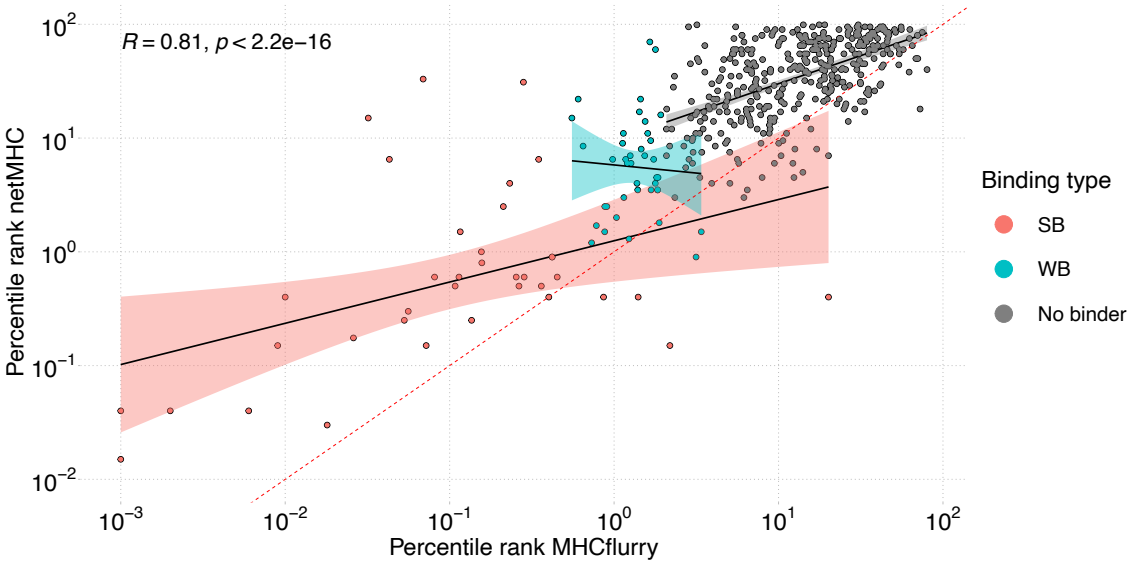
In general, it could be observed that values estimated by netMHC tended to higher levels of Kd (cluster of red dots in the left upper part and cluster of green dots in the middle upper part of Fig. 3.26(b)). To check on a possible association of the concordance of the two prediction tools with the incidence of an HLA-allele, all 92 pairings that satisfied the constraint for weak or strong binders for at least one of the two prediction algorithms were displayed together with their corresponding allele frequencies in a large German population cohort (n=39689; see `http://www.allelefrequencies.net/`; Fig. 3.27).

The linear regression line ($R = 0.62, p < 1.5 * 10^{-9}$, 95%-interval grey shaded) indicated a positive, although again slightly diminished, Pearson correlation for this set of pairings. A humble tendency for more frequent alleles (brighter dots) to exhibit less divergence could be observed. There was no clustering, or qualitative difference between frequently and less frequently expressed HLA alleles.

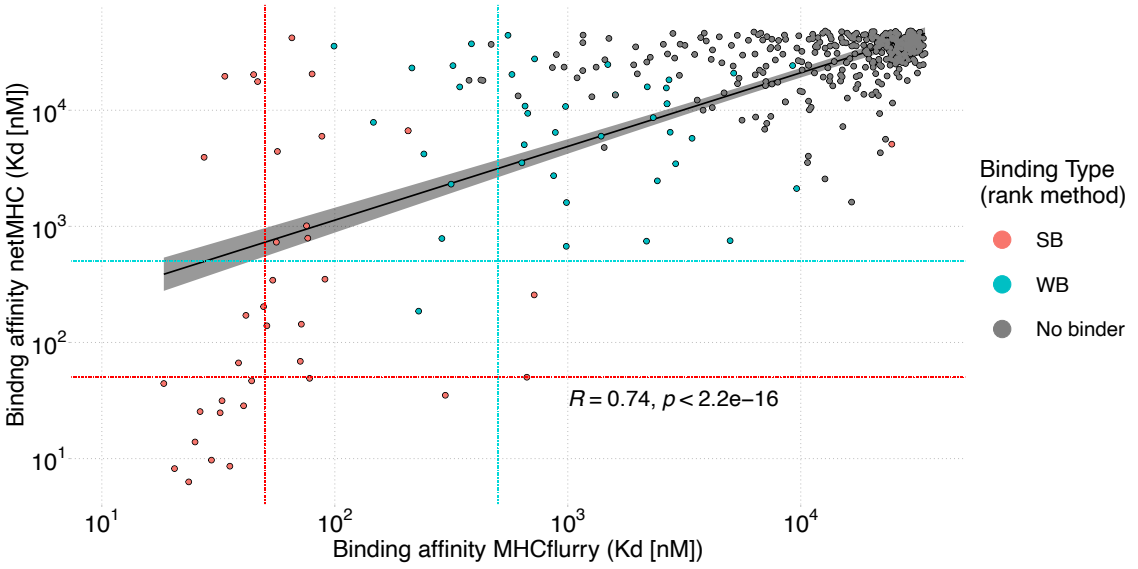
The use of binding affinity predictions as a selection criterion for NACs yet required a unique and comparable value for ranking the different sequences of interest. Hence the most feasible and self-evident way was to use the HLA-allele showing the best value (i.e., the lowest value for K_d or percentile rank) since this represented the setting where MHC class I antigen presentation was most likely to happen for the respective NAC. However, with this approach, other alleles of the same patient were not considered for selection anymore, implicating that even slightly lower affinity values could lead to an allele rejection. On the other hand, a direct affiliation between NACs and HLA-allele significantly simplified the procedures in the subsequent cell-culture experiments and hence may be advantageous in terms of feasibility.
The complete list of NACs with the corresponding best binding HLA-alleles and their estimated binding affinities can be found in Appendix B, 2.1.

3.1.4 Specifications of Neoantigen candidates

Based on the data set for all identified variants (see Section 3.1.2) as well as on the proteogenomic data (Tretter et al., 2023) an algorithm implementing various software tools



(a) Percentile rank



(b) Binding affinity

Figure 3.26: **Pearson correlation analysis between both prediction algorithms.** Each dot represents a NAC-HLA-allele pairing. The results of the predictions of netMHC are plotted vs. those from MHCflurry. The binding type classification was done according to the constraints (see Tbl. 2.15) for the rank method in both cases. The Pearson correlation coefficient R and p-value are shown together with the linear regression lines, which in **(a)** is fitted for each binding type individually (red: Strong binder, green: Weak binder, black: No binder) and in **(b)** is fitted for all data points together (black line). Their 95% confidence intervals are illustrated accordingly. **(a)**, The dashed, red line corresponds to $x = y$. **(b)**, The dash-dotted green and red horizontal and vertical lines illustrate the thresholds for weak (green) and strong (red) binders. NACs with only one prediction available are not shown.

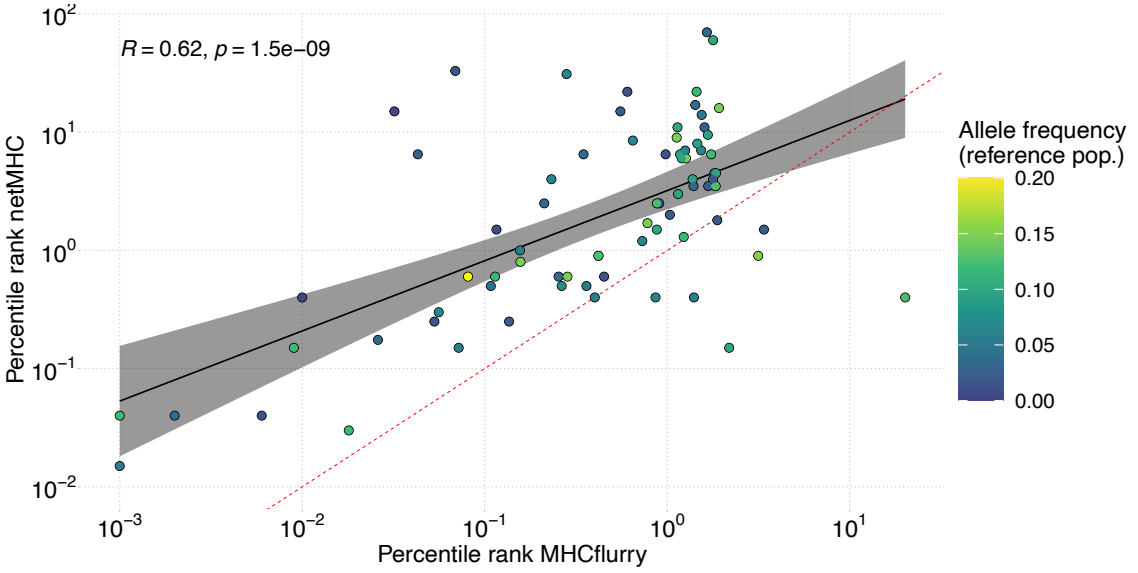


Figure 3.27: **Comparison of the divergence between both prediction algorithms and the corresponding MHC allele frequency.** The percentile rank of all NAC-HLA-allele parings classified as weak or strong binder (for at least one of both algorithms) are displayed for both prediction algorithms together with their linear regression curve (black line) and the corresponding 95%-interval (grey shaded). The color of the dots represents the known HLA-allele frequency in a German reference population. The dashed, red line corresponds to $x = y$.

(*pFind*, *Prosit*; for details see Section 2.2 and Section 3.1.1) was used to integrate the data and obtain a dataset with all possible neoantigens (NACs). Since mass spectrometry facilitates selection for naturally presented peptides, this method was favored over purely binding prediction based epitope discovery methods.
The data containing sequences of all 94 identified NACs and further peptide and origin-specific, as well as patient-associated information, will be discussed in this section.

3.1.4.1 Peptide Candidates from DNA and RNA variants

Besides containing the peptide sequences, the NAC dataset inherits some of the quantities already included in the dataset of variants (see Section 3.1.2.2), allowing to correlate the peptides of interest not only to clinical data, such as tumor entity and metastatic site but also to the genetic origin, the immunopeptidomic calling method and the level of detection of the associated variant.

As illustrated in Fig. 3.28, the major part (N=83) of all NACs resulted from RNA discov-

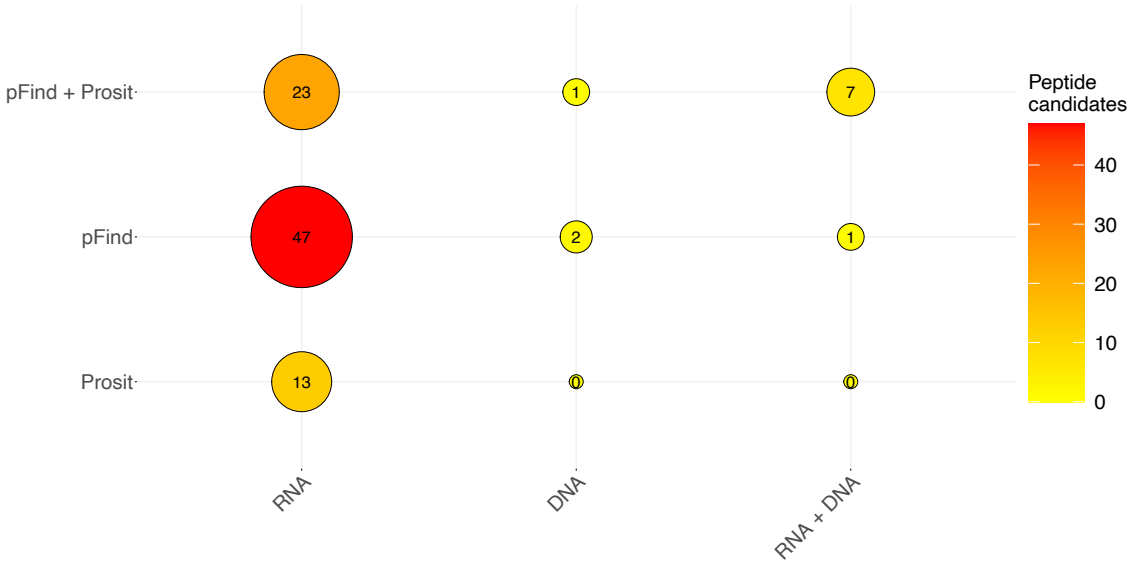


Figure 3.28: **Balloon plot to illustrate the number of NACs (peptide candidates) identified by the different immunopeptidomic calling algorithms and for different sources.** The color and size of the dots represent the number of NACs identified for a specific calling algorithm and a source.

ery, only three candidates emerged solely from DNA variants, and eight NACs could be found in both sources. Again, considering the fraction of NACs identified on RNA level, the vast majority (N=47) could be found with both calling algorithms *pFind* and *Prosit*, whereas 36 NACs were detected either with *pFind* or with *Prosit*. Peptide candidates identified on exome and RNA level seemed predominantly to be found only by the tool *pFind*.

As discussed above (cp. Section 3.1.3), the tumor **VF!** was again used to assess the group of NACs. Considering the subgroups of NACs according to their level of detection, it could be seen that the mean tumor **VF!** was roughly double for RNA-identified peptides compared to those identified on exome level (see Fig. 3.29). By classifying into "inliers" and "outliers" according to criteria defined in section 3.1.2.1, it became evident that the variance of the tumor **VF!** within the group of outlier peptides was significantly higher than that of the group of inliers.

To further evaluate the NACs with regard to possible in-vitro immunogenicity, the results from different binding affinity estimations were assessed. Depicting the predictions obtained with *mhcflurry* (cp. Fig 3.22) revealed that all 94 peptide candidates underlie a widespread distribution with a dissociation constant ranging from >10.000 to far be-

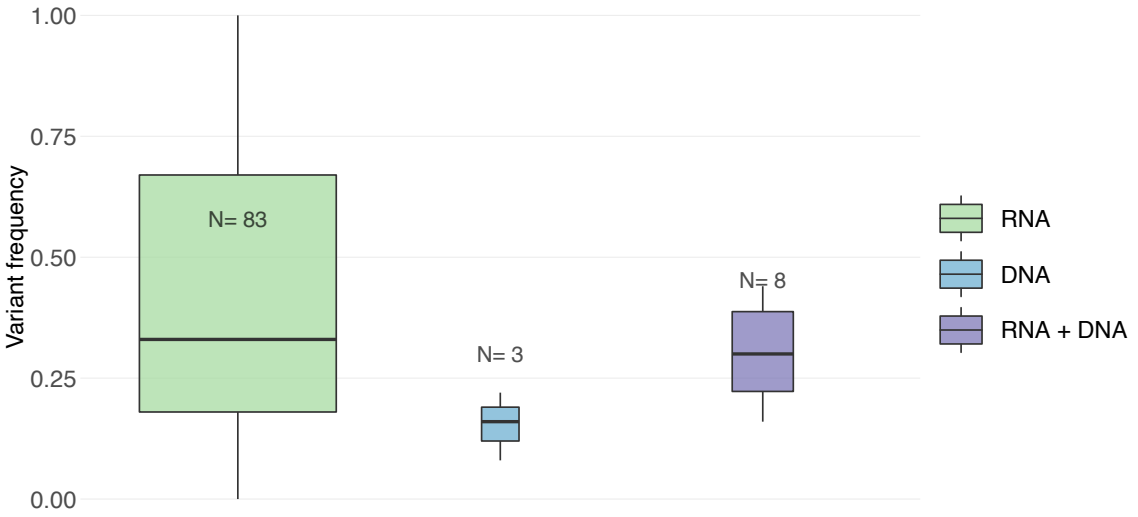
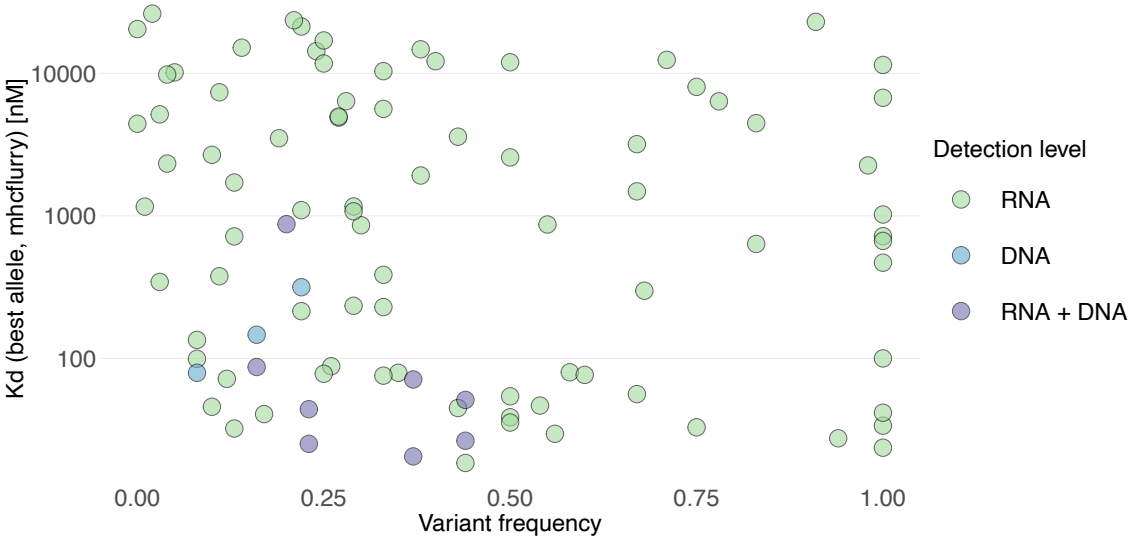


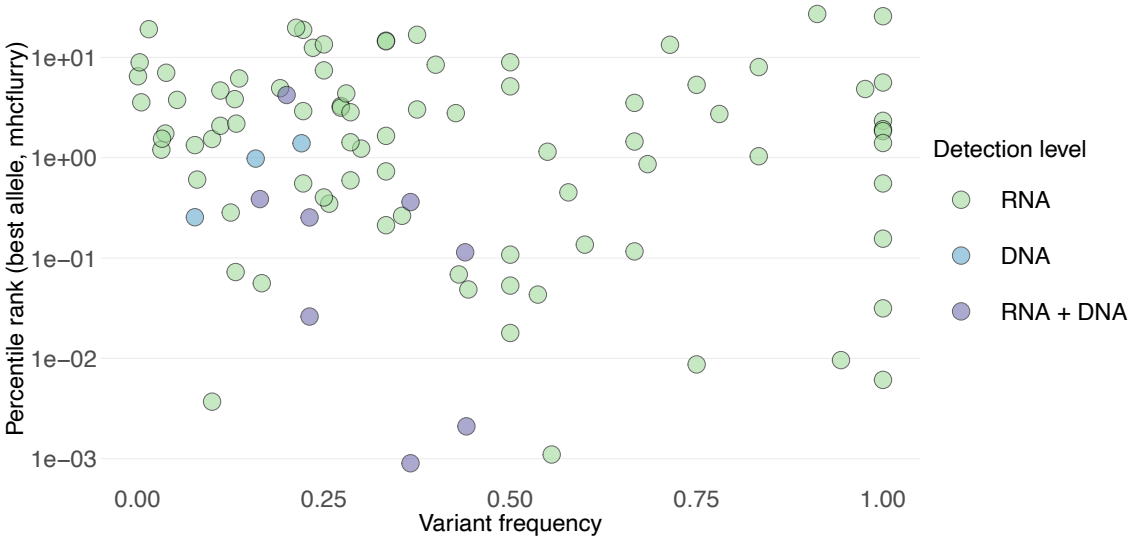
Figure 3.29: **Distribution of the tumor VF of the identified variant forming the basis of the NACs for both DNA and RNA.** The distribution of the tumor variant frequency of all identified NACs is illustrated through boxplots, comprising each DNA-only (blue), RNA-only (green), and variants found on both levels (violet), respectively. A black horizontal line represents the median of each distribution, and the number N of peptides of each group is annotated.

low 100 (see Fig. 3.30). Further discrimination according to the level of detection into peptides identified on exome and transcriptome level led to the striking observation that seven out of eight NACs identified on both regimes (DNA and RNA) were estimated to have Kd<100, which is generally associated to a very high binding affinity. This then led to the assumption that an agreement of results for different detection sources may increase the specificity of the method. More, it could be seen that "DNA-only"-peptides also underlay this clustering effect to some degree. Interestingly none of these eleven NACs detected on DNA level yielded variant frequencies above 0.5.

A similar but less unambiguous effect was observed by highlighting the proteomic tool. NACs that were identified by both tools, pFind and Prosit, seemed to cluster more likely in the regime of low Kd (red dots in Fig. 3.31). In contrast, NACs found only by one of the MS tools (green and blue dots) seemed to be more uniformly distributed over all regimes of Kd and hence did not exhibit a tendency towards higher binding affinities.

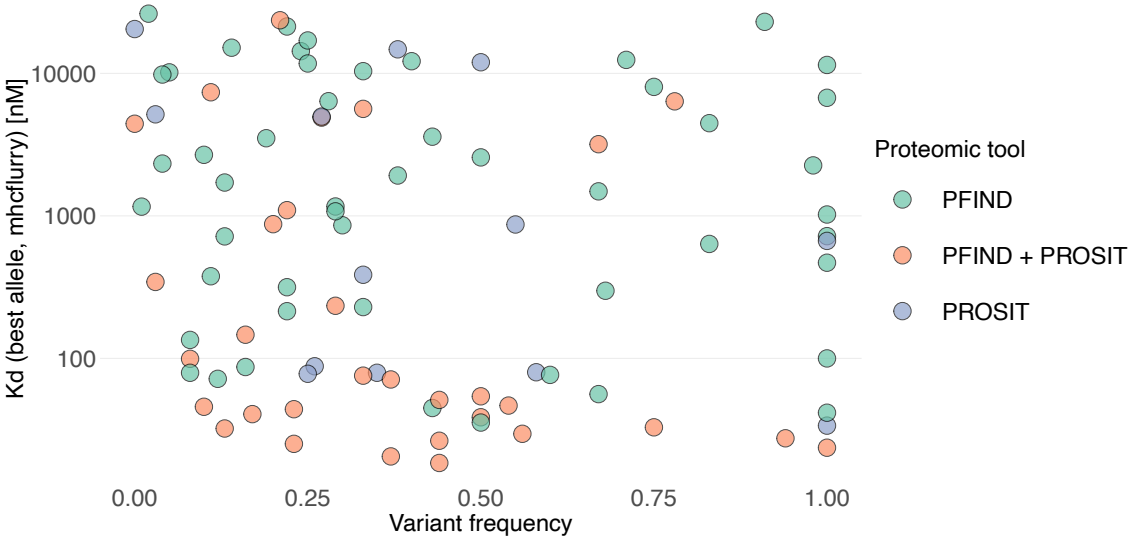


(a) Binding affinity

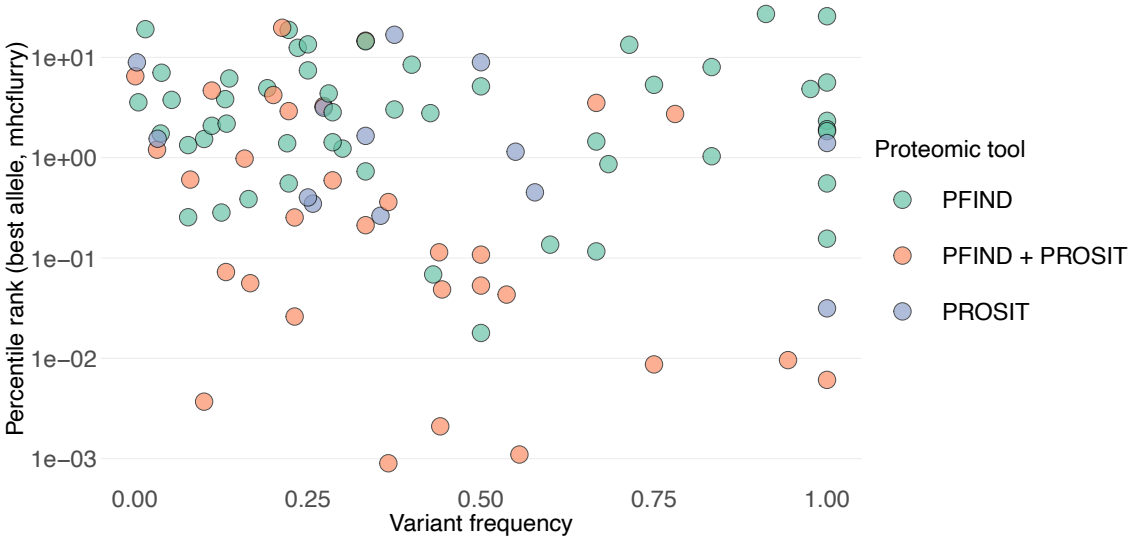


(b) Percentile Rank

Figure 3.30: **Assessment of NACs according to the prediction of their binding properties and the underlying tumor VF with their associated detection level. (a)** The binding affinity (Kd) or **(b)** the percentile rank is displayed versus the tumor VF for every identified NAC.

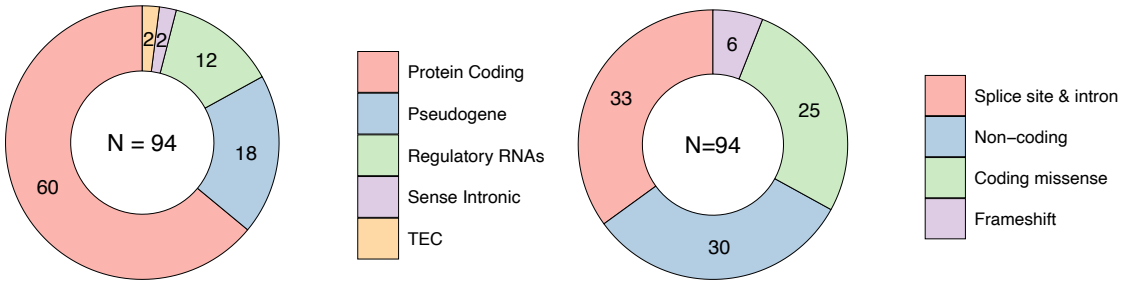


(a) Binding affinity



(b) Percentile Rank

Figure 3.31: **Assessment of NACs according to the prediction of their binding properties and the underlying tumor VF with their associated Proteomic tool. (a)** The binding affinity (Kd) or **(b)** the percentile rank is displayed versus the tumor VF for every identified NAC.



(a) Genetic biotype (b) Variant type

Figure 3.32: **Genetic assessment of NAC underlying variants. (a)**, The genetic biotype of the 94 NACs are illustrated according to their corresponding fraction. **(b)**, variant type assessment for all NACs.

3.1.4.2 Classification and genetic assessment of NACs

The assessment of the genetic origin revealed that the vast majority of all identified NACs (approx. 64%) resulted from variants in protein-coding regions (Fig. 3.32(a)). This was found to be in good agreement with the prior results from the genetic biotype assessment of all variants, which showed a fraction of 78% for DNA and a fraction of 54% for RNA-based variants (cp. section 3.1.2.3). Pseudogenes and regulatory RNAs were found to be responsible for nearly all the rest of all identified NACs (19.1% and 12.8%, respectively).

By repeating the variant type analysis for all variants associated with one of the identified NACs (cp. section 3.1.2.3), an interesting shift could be observed. Whereas beforehand, the major part of all variants was found to be either non-coding or coding missense (DNA: 86%, RNA: 76%; see Fig. 3.12), this fraction was observed to be significantly diminished to 58% for those variants associated to the group of NACs. Furthermore, NACs from splice site and intron variants now (with 35%) even represented the major part (non-codings 32%, coding missenses 27%, see Fig. 3.32(b)). In contrast, for all variants, this fraction was found to be only 2% for DNA and 20% for RNA variants. Only six out of 94 NACs (6.4%) resulted from frameshift mutations, which still implied a slightly increased fraction compared to the group of all variants (DNA: 6.3%, RNA: 2.2%).

3.1.4.3 NAC distribution within the cohort

Considering all cohort patients, in 24 patients, one or more NACs could be identified. The number of potential neoantigens fluctuated from 1 to 14 peptides (see Fig. 3.33) with an

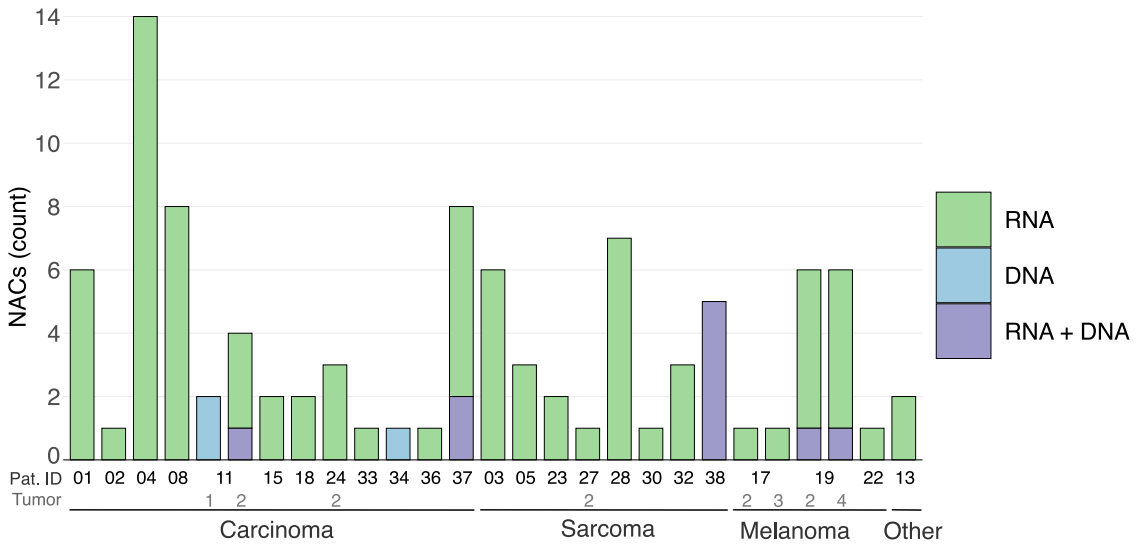


Figure 3.33: **Total number of NACs identified on DNA and RNA level.** The total number of identified NACs is displayed for every patient grouped by four different entity subsets. The color indicates the underlying detection source for every identified NAC.

average of 3.9 NACs per patient. Whereas in some patients, respectively tumor samples, all NAC-discovery is due to identification on DNA level (i.e., no available RNA-data as in patient 34 and tumor one from patient 11; see Fig. 3.33) most of the patients showed at least one peptide based on variants detected on RNA level. In some patients with multiple metastases, the yield of NAC was similar in both tumors (patient 19; tumor 2 and 4), while others showed a more heterogenous structure (patient 24, only NACs identified in tumor 2).

3.1.4.4 High affinity MHC alleles

Since immunogenic responses of reactive $CD8^+$ T cells that were investigated in this work mainly resulted from neoantigens presented by MHC class I, a closer look at the distribution of HLA alleles was necessary. Here all three loci, "HLA-A", "HLA-B", and "HLA-C", were considered independently. Comparison of the different allele frequencies within the cohort with those of the German reference population (Gonzalez-Galarza et al., 2019) led to some interesting observations.

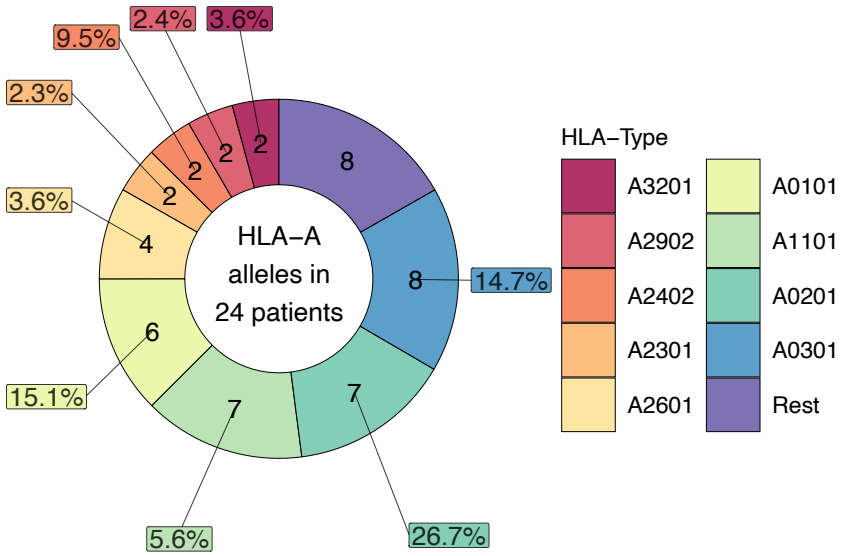
The HLA allele with the overall highest frequency in the cohort was found to be HLA-A03:01 (n=8), which was detected in one-third of all patients. This allele has only a fre-

quency of 14.7% within the German reference population (see Fig. 3.34(a)). Whereas the most common allele within the German reference population (HLA-A02:01) seemed to be roughly as frequent as in the cohort compared to the reference population (29% vs. 26.7%), the allele HLA-A11:01 exhibited a relatively high frequency in the cohort (n=7). It could be detected in about 29% of the cohort's patients, but less than 6 % of the German population is a carrier of this allele. Similar observations could be made for HLA-B and HLA-C epitopes, where very rare HLA-types could be found in a significant part of the cohort (i.e., HLA-B37:01, HLA-B40:02, and HLA-C01:02; see Fig. 3.34(b) and 3.34(c)).

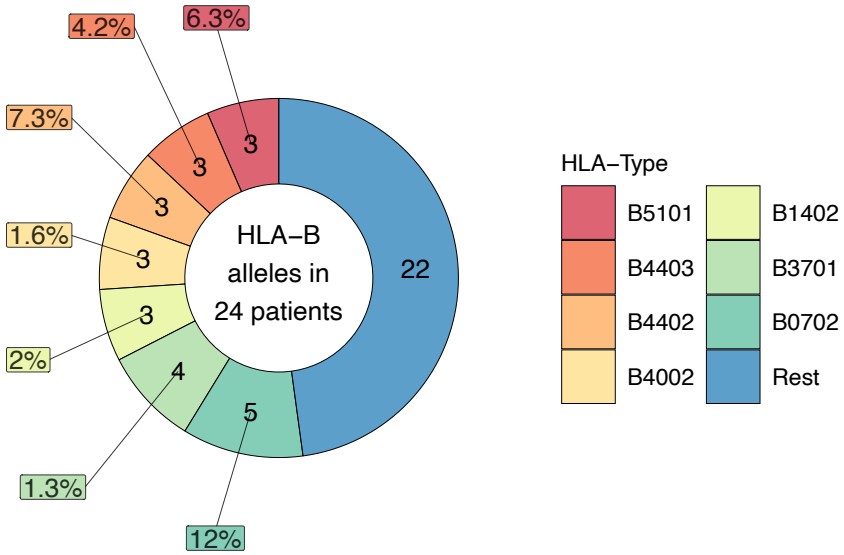
By performing in silico predictions of binding affinities (see Section 2.2.3 and 3.1.3.1) for all possible MHC class I - neoantigen combinations, and classification of all available alleles for one specific peptide, different observations could be made.

First, all three HLA subtypes (HLA-A, HLA-B, HLA-C) were similarly likely to be the most probable binding partner (per average). Hence in terms of numbers, no clear tendency could be observed. Second, different specific HLA alleles differed significantly in terms of binding affinity, although they were identified comparably often as the best-binding allele for a group of peptides. For example, the allele A11:01 was found to be the best binding allele for seven peptides within six patients (Fig. 3.35(a)). With one exception, they all yielded a binding affinity in the strong binding regime. In comparison for the allele C06:02, which was found to be the best binding allele in eight cases (five patients), only one yielded a suitable binding affinity that did not classify in the non-binding regime.

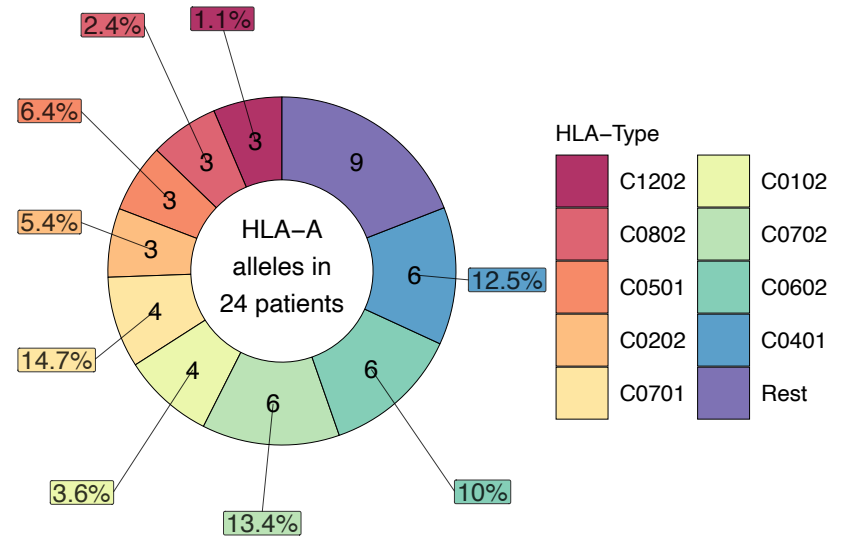
Third, both ranking parameters (lowest dissociation constant vs. lowest percentile rank) were compared head-to-head. Kd-based ranking favors a preferential predicted binding to HLA-C, whereas analysis of the percentile rank seems to favor binding an HLA-A or HLA-B allele (see Fig. 3.35(b)). However, most of these peptides belonged to the low-affinity group, i.e., non-binder. The results for peptides that were most likely to bind to an HLA-B allele not only appeared to be more stable considering both prediction values but also showed a greater HLA variability, meaning that there were more different HLA-B subtypes (17 vs. nine for HLA-A, cp. Fig. 3.35(b)), that were identified as best binding HLA allele.



(a) HLA-A

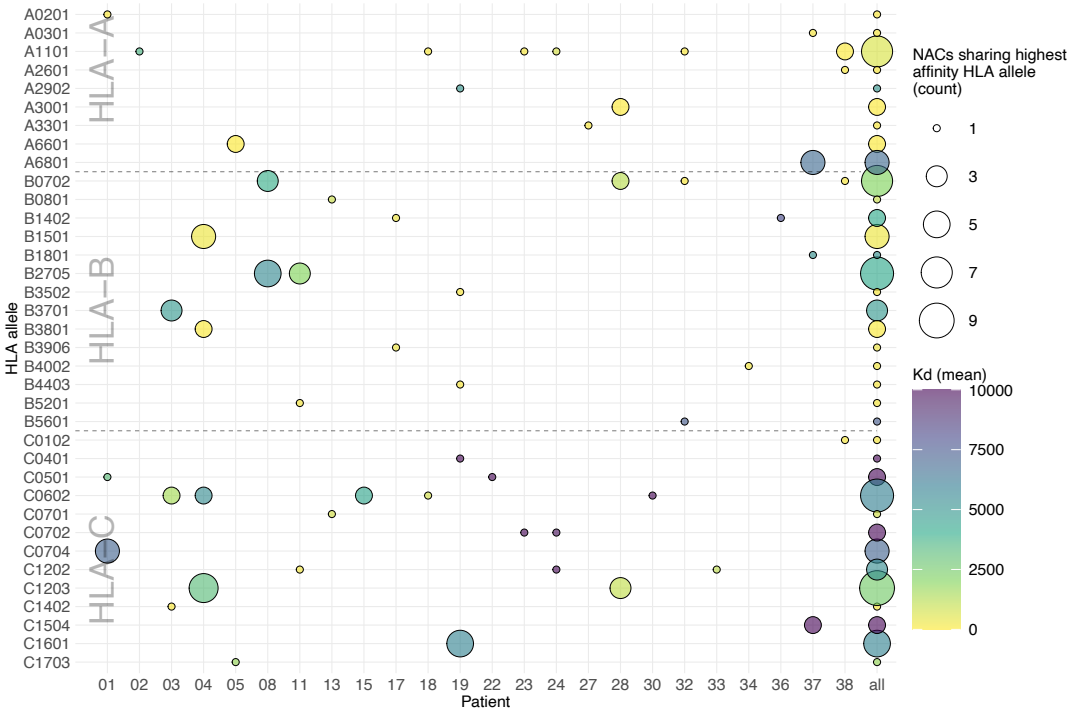


(b) HLA-B

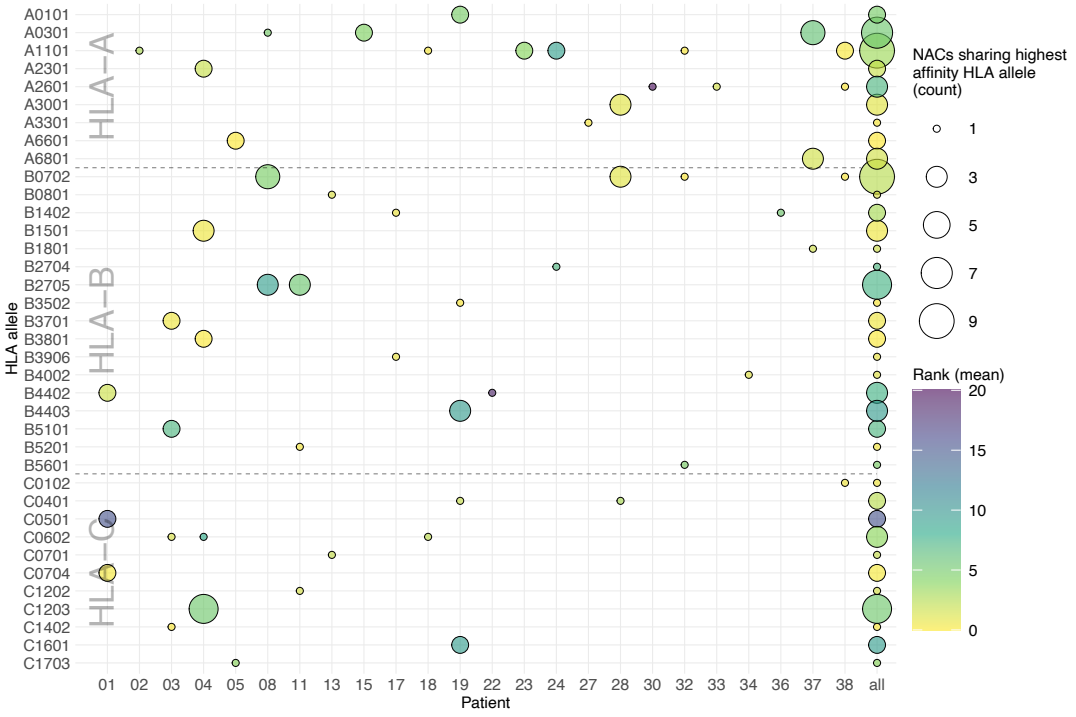


(c) HLA-C

Figure 3.34: **Distribution of HLA types within the cohort.** The additional labels refer to the according HLA allele frequencies in the reference population.



(a) According to the binding affinity (Kd)



(b) According to the percentile rank

Figure 3.35: **Best binding HLA-alleles for all NACs displayed for each patient.** For all possible combinations of HLA-allele and patient, the corresponding number of NACs is displayed (size of the dot) together with the mean binding affinity **(a)** and the percentile rank **(b)** (color of the dot). HLA alleles from one of the three HLA loci affiliated with the MHC class I are depicted together, separated by a dashed line. The last column represents the sum of all patients (size of the dot) and the mean of all individual affinity values assigned to this HLA-allele over the whole cohort (color of the dot).

3.1.4.5 Assessment of mutational variability of binding affinities

Out of all 94 identified NACs, 51 contained a somatic mutation localized within the peptide sequence and hence could be associated with their corresponding **WT!** (**WT!**) peptide. In case of an actual presentation of the **WT!** peptide, the NAC could hence be competing for the same binding sites at the appropriate MHC class I molecule. A comparison of both predicted binding affinities, the **WT!** sequence and the sequence with a single somatic mutation, may help to estimate the likelihood of MHC class I neoantigen presentation. Thus, the affinity change from **WT!** to mutated peptide was assessed.

As an indicator of the change in affinity, the difference in the logarithmic rank of both peptides was used

$$\Delta_{\log(rank)} = \log 10(rank(peptide_{mutated})) - \log 10(rank(peptide_{WT})) \tag{3.1}$$

$\Delta_{\log(rank)}$ gives the change in the order of magnitude of the estimated rank for each NAC/WT couple.

34 out of 51 NACs were subject to an increased estimated binding affinity (reduced rank, reduced $\Delta_{\log(rank)}$). In contrast, for 14 NACs, a lower binding affinity was observed (see Fig. 3.36). In three cases, there was no difference at all. The average (mean) for $\Delta_{\log(rank)}$ was observed to be about -0.22. This implied that among all identified NACs with a somatic mutation within the peptide sequence, the majority tended to exhibit an increased affinity compared to the WT peptide. Classifying the binding types for the wildtype and the mutated peptides showed that two NACs, for which the corresponding WT peptide had been rated as non-binder, now shifted towards the strong binding value and three towards the weak binding regime (red and orange bars in Fig. 3.36; see Table 2.15 for definition of binding regimes). Four peptides experienced a shift from weak to strong binder (green bars in Fig. 3.36).

Out of 14 NACs that, through the mutation, showed reduced binding properties, only two peptides were down-ranked as weak or non-binders when compared to their corresponding WT sequence. Considering the predicted HLA allele that exhibited the highest binding affinity, five cases were holding a lower $\Delta_{\log(rank)}$ by interchanging the present allele for the mutated version of the peptide (see Fig. 3.36). Four of these five were asso-

ciated with an increased overall binding affinity, whereas one showed a lower affinity.

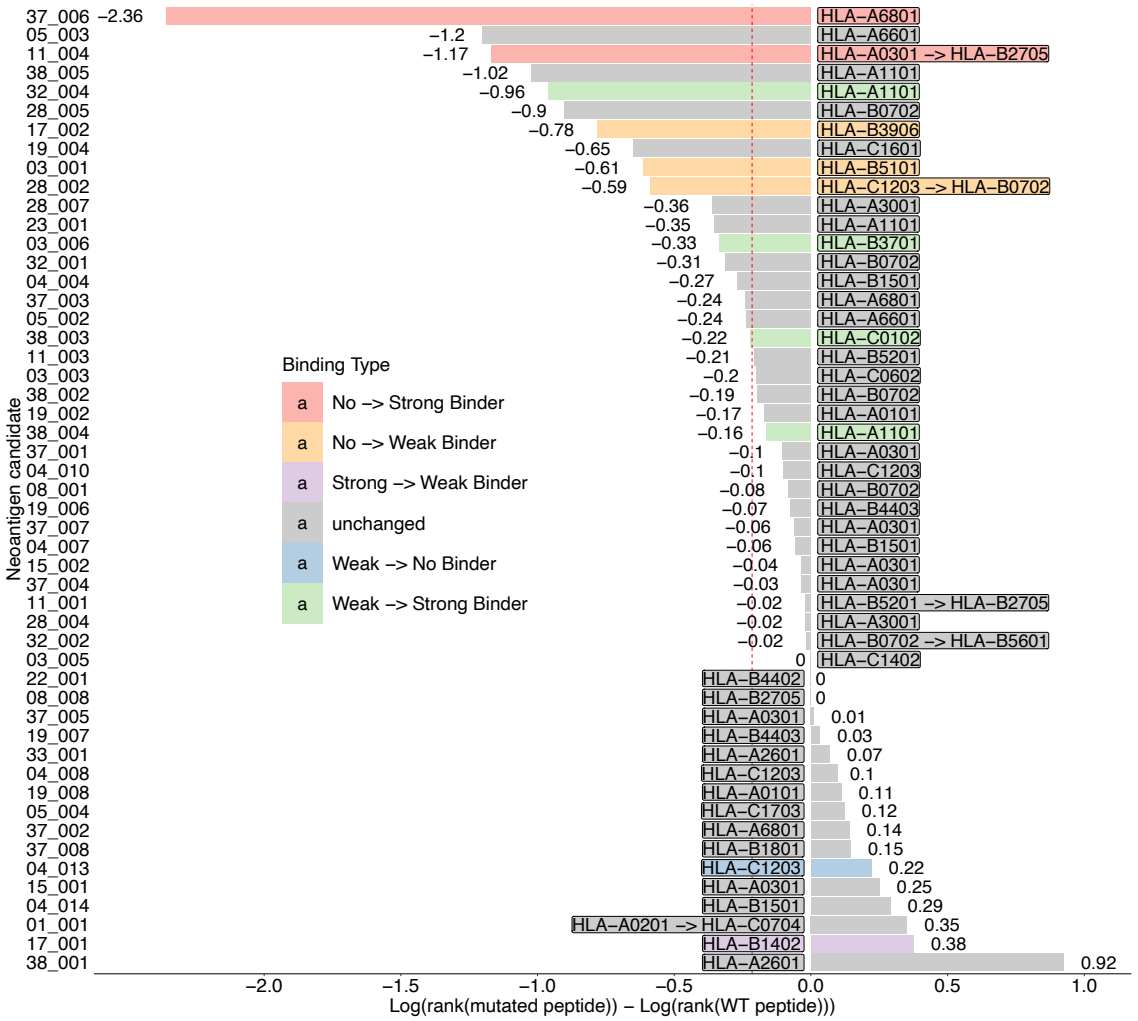


Figure 3.36: **Binding affinity alteration (rank) of WT-peptide compared to NACs** The change of the binding affinity of NACs compared to their associated WT-Sequence is depicted. For better visualization, the log value of the percentile rank for both sequences was selected. The annotated value corresponds to the Delta of the order of magnitude of the rank, where negative values are associated with a decreased rank and hence indicate an increased binding affinity. The red dotted line indicates the average (mean) shift over all peptides.

3.2 In-vitro stimulation of T cells

The assessment of immunogenicity of possible neoantigens (NACs) was done with an accelerated Dendritic cell culture (acDC) involving T cells from patient blood samples (i.e., from patient PBMCs, for details cp. Sec. 2.3.2). Until the writing of this thesis, 57 different NACs were tested within 192 assays with samples from eight different patients. However, due to continuous modification of different bioinformatical procedures, a major part of the already assessed NACs had to be deprecated retroactively. This was due to an updated control database that led to the identification of 41 WT-peptides out of the 57 initially identified NACs. These peptides were hence considered in a separate section (see 3.2.2). The other 37 NACs were not assessed for one of the following reasons: They were either deprecated before being tested, there was insufficient patient material for testing, or the synthesizing process of the peptides could not be performed successfully.

3.2.1 Assessment of actual NACs

At the moment of the writing of this thesis, 16 out of 57 tested NACs were still listed as potential neoantigens and are designated as "actual NACs" in the following. Thus, the results of 26 assays were assessed for immunogenicity and are shown in this section. The complete list of actual NACs can be found in Appendix B, 2.1.

To investigate NAC immunogenicity, the amount of T-cell reactivity for different conditions had to be compared. Here, two conditions were evaluated, namely pulsing with the NAC of interest and pulsing with a WT or irrelevant peptide, respectively, and the corresponding number of spot forming units (SFUs) was assessed.

The final immunogenicity assessment of all realized assays was then done based on two quantities. First,

$$\Delta_{SFU} = N_{SFU,\ AG-pulsed} - N_{SFU,\ irP-pulsed} \tag{3.2}$$

where $N_{SFU,\ AG-pulsed}$ and $N_{SFU,\ irP-pulsed}$ are the mean values of the number of SFUs for the antigen-pulsed and irrelevant peptide-pulsed condition, respectively. Second,

$$R_{SFU,\ rel} = \frac{N_{SFU,\ AG-pulsed}}{N_{SFU,\ irP-pulsed}}. \tag{3.3}$$

For NACs to be considered as "immunogenic", Δ_{SFU} had to be greater than 50, and $R_{SFU,\,rel}$ had to be greater than two.

Of the actual NACs, none fulfilled these criteria (Fig. 3.37(a)).

Looking again at the binding affinities of the assessed NACs, no clustering or obvious correlation between Δ_{SFU} or $R_{SFU,\,rel}$ and the predicted affinities could be detected (see Fig. 3.37(b)).

3.2.2 Assessment of deprecated NACs

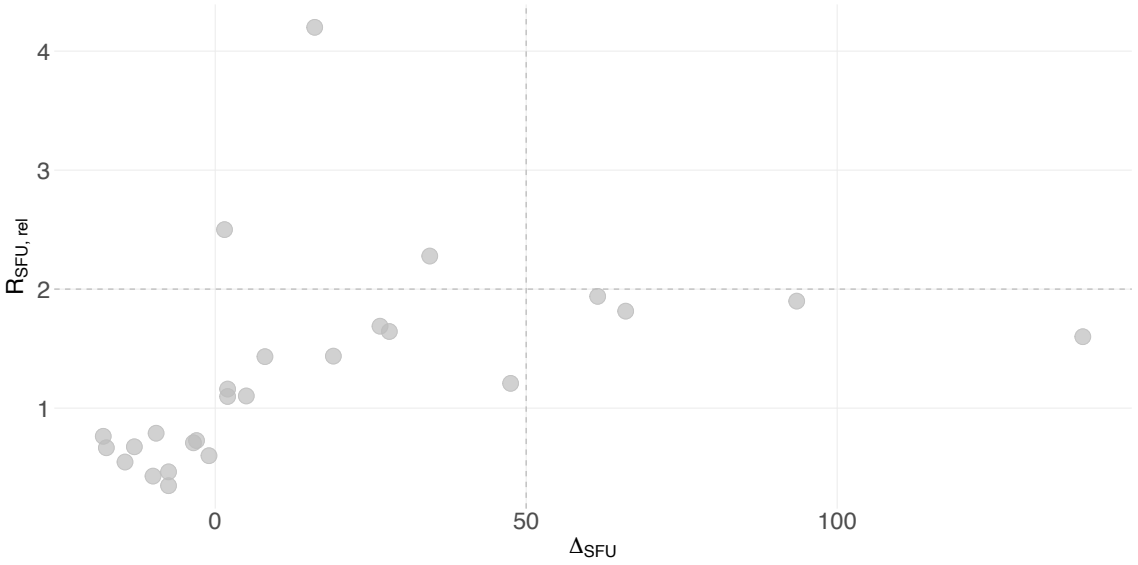
This section describes the experimental results of the remaining 41 NACs. Since they were no longer listed as actual neoantigen candidates due to pipeline modifications, they were labeled as "deprecated" in the following. Most deprecated peptides were found in patients 1, 11, and 19.

The complete list of deprecated NACs can be found in Appendix B, 2.2.

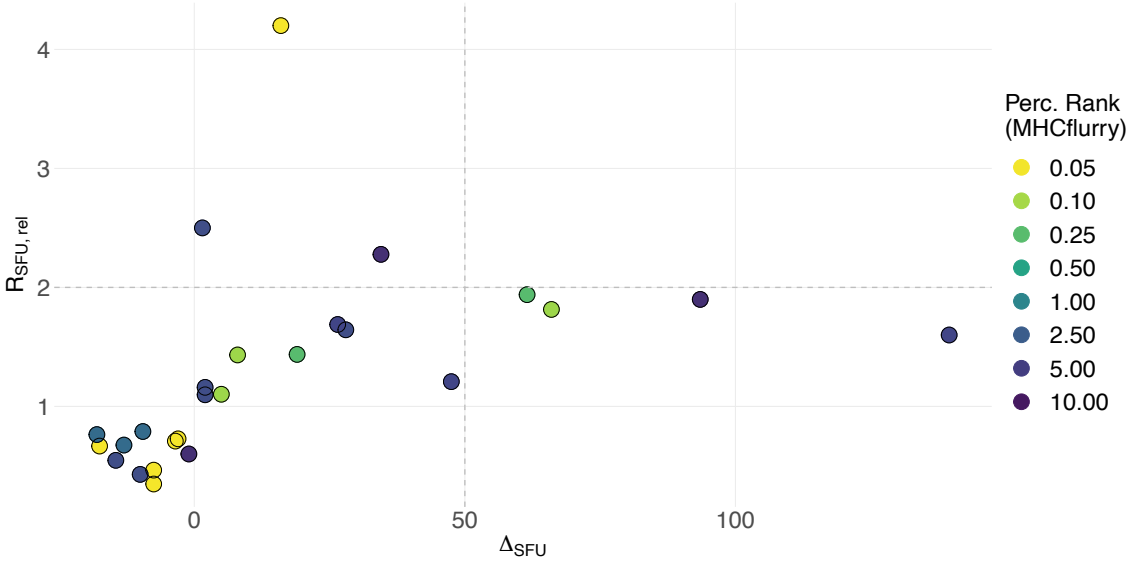
Again all deprecated NACs were analyzed based on the two conditions "antigen-pulsed" and "irrelevant-pulsed" according to Equations 3.2 and 3.3.

Considering the same thresholds as before, three of 116 measurements (2.6%) for three different NACs were found to meet the criteria for immunogenicity (see Fig. 3.38(a)). The data points showed a large spread in Δ_{SFU} (50 to 162) and a relatively narrow spread in $R_{SFU,\,rel}$ (2.1 to 3.5). The identified peptides resulted from patients 1, 11, and 19.

Plotting the binding affinity (percentile rank) of the identified peptides indicates a vague positive correlation trend between Δ_{SFU} and the predicted binding affinity. (Fig. 3.38(b)).

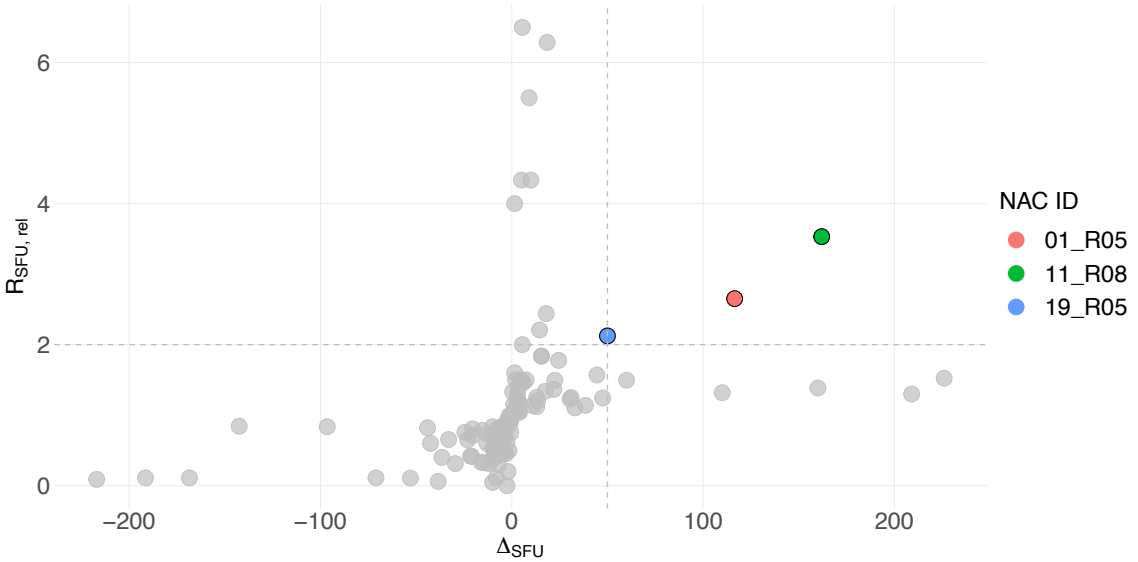


(a) Immunogenic NACs (highlighted)

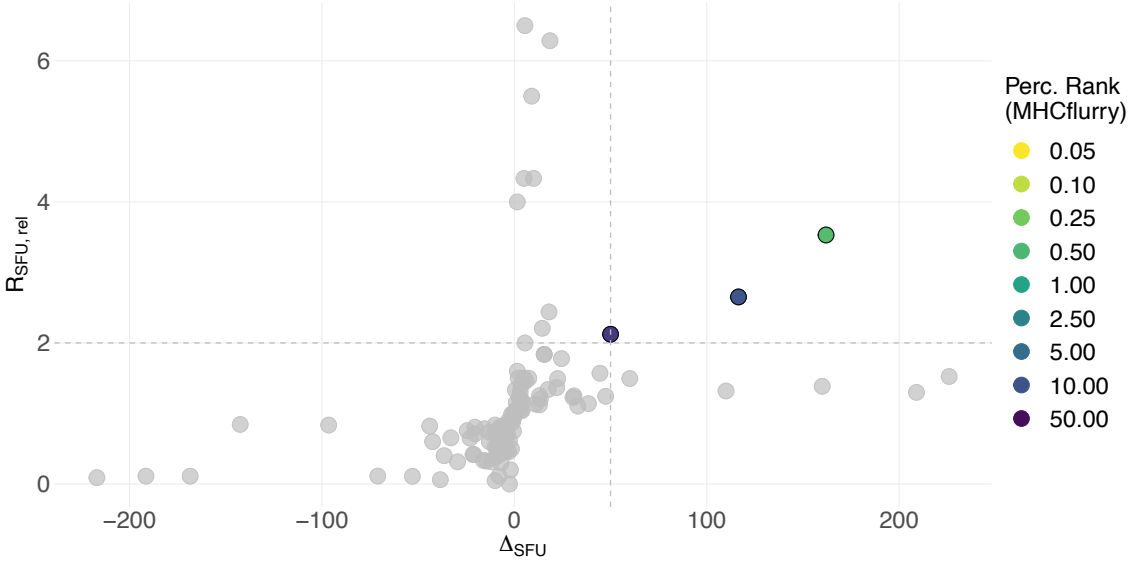


(b) Affinity assessment of all immunogenic NACs

Figure 3.37: **Assessment of Immunogenicity of actual NACs with two conditions.** For all tested NACs Δ_{SFU} is displayed versus $R_{SFU,\,rel}$. **(a)**, All data points meeting the conditions for immunogenicity are highlighted (with colors), and the corresponding thresholds are depicted with dashed grey lines. The remaining data points are illustrated with light grey dots. **(b)**, The corresponding best percentile rank is shown for all actual and immunogenic NACs within the same representation. Lighter colors represent higher binding affinities. *The shown data has been produced together with Celina Tretter. It was in parts published by Tretter et al. (2023)*



(a) Immunogenic NACs (highlighted)



(b) Affinity assessment of all immunogenic NACs

Figure 3.38: **Assessment of Immunogenicity of deprecated NACs with two conditions.** For all deprecated but tested NACs Δ_{SFU} is displayed versus $R_{SFU,\,rel}$. **(a)**, All data points meeting the conditions for immunogenicity are highlighted (with colors), and the corresponding thresholds are depicted with dashed grey lines. The remaining data points are illustrated with light grey dots. **(b)**, The corresponding best percentile rank is shown for all deprecated and immunogenic NACs within the same representation. Lighter colors represent higher binding affinities. *The shown data has been produced together with Celina Tretter.*

4 Discussion

4.1 Preliminary remark

The ImmuNeo project is composed of different sub-projects collaborating by continuous exchange of data with subsequent modifications of workflows. For this reason, the pipeline and all described results in this work depict an in-process project status, thereby centering on some aspects of a scientific process underlying permanent improvement and modification at different levels. Thus, there is no claim, neither for completeness nor for the up-to-date nature of the presented data.

4.2 Mutanome variations

4.2.1 Patient and disease dependent fluctuations in the genetic landscape

Since only a small fraction of somatic non-synonymous mutations (NSMs) are represented as neoantigens that may serve as immunogenic targets for therapy (Garcia-Garijo et al., 2019), a reliable and acceptably comprehensive analysis of the mutanome was required. However, the genetic landscape between different patients varied significantly. Especially for DNA, the number of genetic variants fluctuated tremendously, leading to mutanome size variations between 20 variants for patient 16 and 750 variants for patient 38. The final NACs outcome was observed to correlate with the number of DNA and RNA variants, indicating that a high number of variants (i.e., as for patient 38) would have an increased likelihood for significantly higher numbers of detected NACs.

The huge variety of DNA and RNA variants found in the data can be explained partially by a high variation of tumor mutational burden (TMB) across tumor types (Chan et al.,

2019). Disease entity-specific mappings of these variations indicate that the number of variants per bp found in melanoma samples may be 10 to 100-times higher than in low-TMB tumor entities, which is in good agreement with the mutanome data of ImmuNeo patients (Chalmers et al., 2017; Zehir et al., 2017). However, a significant spread of the distribution is also observed within the same cancer type (Chan et al., 2019), which can be explained by different tumor-specific factors (such as UV-light and smoking) causing tumor somatic mutations responsible for certain entities (Alexandrov et al., 2013b). For the ImmuNeo data set, these variations range from roughly 120 to 500 DNA variants within the group of melanoma patients, which is also in good agreement with the data described in above mentioned literature.

The tumors of patient 11 and patient 38 inherited markedly higher numbers of variants than the rest of the cohort. For patient 11, a mismatch repair deficiency (MMRD) was diagnosed, which is known to be associated with a significantly increased TMB (Grant et al., 2021). However, the corresponding RNA variant count (data availability only for tumor two) did not exhibit high values compared to the rest of the cohort. With a DNA variant count above 750, the TMB of patient 38 exceeded all other investigated tumor samples. Considering that a recent publication describes malignant peripheral nerve sheath tumor (MPNST)-like melanomas as high TMB entities (Pimentel Muniz et al., 2020), this might be an interesting finding. MPNST-like melanomas could thus shape up a tumor entity with above-average responses to immunotherapeutic treatments.

In comparison, a less homogeneous distribution for variant counts on RNA level (except for melanoma) may indicate that also other factors than the TMB could explain observed data. Such other parameters may also be causal for unexpected high or low DNA variant counts. First, data acquisition related factors have to be mentioned. Here especially, the size of the analyzed tumor tissue and the intra- and inter-tumoral heterogeneity are of outstanding importance. They may directly affect the number of possibly detectable variants (Litchfield et al., 2020; Shi et al., 2018; Jacoby et al., 2015) and directly depend on the anatomical region as well as the size of the resected tumor tissue. The sequencing depth and the quality of the DNA sequencing itself are other parameters potentially influencing the results (Griffith et al., 2015).

When analyzing variants shared between different samples, a statistical validation would require a much higher number of patients within the same tumor entity group. Hence,

assessing a much bigger cohort focusing on specific cancer entities would be demanded to support described findings further.

In summary, low numbers of variants turned out to be a crucial limitation for discovering possible neoantigens in most patients. In this context, the data set did not only reflect canonical somatic mutations on coding exons as well as mutations on non-coding regions. Due to the additional assessment of the transcriptome, different transcriptional and post-transcriptional events were also detected, revealing additional complexity of tumor evolution and immunogenicity. Hence, this has to be considered for all further hypotheses on disease development, progression, drug response, and clinical outcome. Exploiting all possible repertoires of variants seems indispensable for future identification pipelines.

4.2.2 Differences in DNA and RNA variant coverage

By inclusion of RNA-seq data in the identification pipeline, the ImmuNeo project could substantially increase the number of genetic variants and neoantigens, especially for non-melanoma entities. A special methodology combined tumor RNA-seq with normal WES data to exclude false positive RNA variants. This procedure was shown to be the most feasible and effective option for calling RNA variants (Hashimoto et al., 2021).

In the context of NAC validation, a distinct classification of variants is of utmost importance. Here, variants simultaneously detected on exome and transcriptome levels were assumed to have an a priori higher likelihood of an actual presentation, therefore eliciting immunogenicity in vitro. However, the detailed comparison of RNA and DNA variants led to the intriguing observation of a tiny shared fraction of DNA and RNA variants within the ImmuNeo cohort ($\sim$ 1.4% of all detected variants). This value fluctuates significantly considering different entities, with melanoma patients showing the highest overlap ($\sim$ 3.5%) to values below 1% for other entities.

The fraction of DNA variants covered by RNA is relatively stable (between 30% for melanoma and 42% for others, Fig. 3.10), but a vast amount of RNA variants cannot be detected on DNA level (only between 0.6% for others and 3.8% for melanoma). This may underline the importance of RNA-seq to detect variants not occurring at the DNA

level but being the result of RNA processing events like alternative splicing and RNA editing (Eisenberg & Levanon, 2018; Tan et al., 2017; Xu et al., 2018). Thus, these RNA-associated events, as previously reported, may contribute substantially to the diversification of the cancer proteome, qualifying RNA-seq as a valuable additional source of potential neoantigens (Peng et al., 2018; Laumont et al., 2018).

Nonetheless, exploiting the transcriptome to expand the source of possible cancer neoantigens (Zhou et al., 2020) comes with certain limitations. First, bioinformatic algorithms directly influence the variant outcome. As they differ extensively across different gene panel platforms, a certain degree of transparency is needed for setting biologically motivated thresholds, such as cut-offs for variant allele frequencies (Chan et al., 2019). Second, the applied filtering criteria (cp. Sec. 2.2.1.4) led to entirely different exclusion profiles between DNA and RNA, meaning that the distribution of variant reads of the underlying RNA data caused that only one-third of all possible RNA variants were excluded (cp. Fig. 3.7). In comparison, on DNA level, more than half of all variants were rejected due to these criteria, even though there was no a priori evidence for higher FDR in DNA than in RNA data. Further optimization of filtering strategies for RNA compared to DNA here may balance the ratio of false positive data to false negative candidates and help to increase the likelihood of the detection of true neoantigens.

Detailed analysis of dataset-specific parameters of the underlying RNA-seq data revealed a clearly reduced mean sequencing coverage per identified variant compared to DNA data. Both distributions, for DNA and RNA, exhibited different characteristics, with the RNA data indicating a higher variability in coverage and sequencing depth, which could be responsible for an increased artifact yield of RNA-based NACs compared to those based on DNA data.

At the time of this writing, the *Genotype-Tissue Expression (GTex)* database was incorporated into the bioinformatic procedures of the ImmuNeo project to detect and exclude read contamination of RNA-seq data (Nieuwenhuis et al., 2020). Due to these pipeline modifications that were not part of this work and that were hence not considered in the results part, a fraction of the identified RNA variants turned out to be labeled as mutant, although these alterations could be detected in healthy tissues to a varying extent. Applying these considerations to the actual dataset of this work may critically alter results based on RNA variant analysis. Reanalysis of proteomic data with spectral angle analy-

sis and other validation methods were therefore integrated into the actual version of the manuscript accepted for publication (Tretter et al., 2023).

4.2.3 Intermetastatic variant analysis

As shown by previous publications, single-cell sequencing of different clonal subpopulations from different metastasis (and its corresponding primary tumor) provides insight into tumor evolution and pathogenesis (Navin et al., 2011; Jones et al., 2008). The genetic analysis of the different tumor metastases for several ImmuNeo patients reveals not only the overlap size, allowing for estimating the size of the subclonal populations. The results also show a symmetric overlap in the case of patient 19. A well-balanced picture was observed for the three different tumor tissues when considering the distribution and intersections of variants. This observation supports the notion that further information about tumor evolution might be concealed in this data. Further in-depth analysis of sequencing data with methods such as "lineage-tracing" could even guide future therapeutic strategies, as they indicate that metastasis-private mutations do not seem to be drivers of cancer spread but might be associated with drug resistance (Gui & Bivona, 2022; Hu et al., 2020).

Again, in the context of this inter-metastatic variant analysis, the results for melanoma patients have to be highlighted. Here the mutual variant overlap exceeds those of all other assessed entities (Fig. 3.21). A potential explanation for this remarkable result could be found by discussing different melanoma-adapted tumor progression models as suggested by Motwani & Eccles (2021). Further inter-metastatic analysis of different tumor entities could not only emphasize and explain the outstanding performance in the immunotherapeutic treatment of melanoma diseases but identify other eligible entities for these treatment modalities.

4.3 HLA binding affinity predictions

4.3.1 In silico binding predictions as selection method for neoantigens

So far, most approaches have used the well-known canonical repertoires, that is, the usage of DNA data as templates for MS spectra matching. In comparison, the realized repertoire extension with transcriptome data comes with a substantially increased number of possible targets. However, a primary experimental evaluation of all emerging peptides would not only be resource-consuming but, as time is critical for most therapy settings, not feasible for routine clinical application (Arnaud et al., 2020). For reasons of technical feasibility, an efficient pipeline for neoantigen discovery hence requires, at any rate, a rigorous selection method that a priori divides potential NACs, that are capable of inducing durable T cell responses from other peptides that are less likely to be an effective target (Zaidi et al., 2020).

Up to this writing, in silico peptide binding prediction algorithms depicts the most frequent option to rank potential NACs for the likelihood of eliciting an immunogenic T cell response (Wells et al., 2020). By this means, in the ImmuNeo project, about one-third to one-half of all NACs could be ranked as potential binders (depending on the used algorithm), and one-half of these were further classified as strong binders (Fig. 3.24). Nevertheless, a comparison of both algorithms indicates that, especially for the intermediate regime (weak binders), comparably lower prediction reliability has to be assumed since a great extent of divergence was observed between both tools. In this regard, it has to be taken into consideration that the key pre-selection of potential targets was realized by the inclusion of MS data providing additional evidence for the actual presentation of respective NACs. Ongoing adjustments and improvements of the used prediction algorithms, together with advancements in the field of immunopeptidomics, may lead to promising future solutions (Conev et al., 2022; Bulik-Sullivan et al., 2019; Chong et al., 2022).

4.3.2 Divergent prediction outcomes based on methods and tools

Besides the general challenge of ranking possible neoantigens for potential immunogenicity, a comparison of the rank method and nano-molar binding affinity prediction has shown major discrepancies when using well-established thresholds from literature (Bonsack et al., 2019). Applying both methods in a parallel approach yielded significantly different results. A direct comparison could hence be used to describe the estimated binding properties of the collectivity of the corresponding HLA-allele in the model. Following the line of argument of Paul et al. (2013), this allele-specific distribution could reflect that different alleles also vary in terms of epitope repertoire size, hence exhibiting better immunogenic properties in general. Alternatively, a variation in the immunogenicity-associated affinity threshold could also cause this effect, indicating that allele-dependent thresholds are necessary for a comparable selection in terms of immuno-response. Furthermore, this observation could be explained through differently trained HLA predictors that are used by binding prediction algorithms (Jurtz et al., 2017; Nielsen & Andreatta, 2016b).

The deviation was also explicit for best binding HLA assignment, where a significantly higher fraction of NACs was assigned to HLA-A and HLA-B alleles when using the percentile rank method. In contrast, using the binding affinity method strongly favored HLA-C alleles. Depending on defined thresholds, this might generally lead to an underrepresentation of specific alleles. As mentioned above, general HLA-independent thresholds for all samples and alleles could be causal for this bias. In contrast, specific allele-adapted thresholds might provide a more accurate picture of peptide binding strengths. As indicated by Fig. 3.25, the deviation is highest for intermediate regimes, where the distance between the sigmoidal binding curves is greatest.

4.4 Specifications of NACs

4.4.1 NAC detection level analysis

In good agreement with the results of the analysis of DNA/RNA variant overlap, where 97% of all variants were identified on RNA and only 4.4% on DNA level, most NACs

(97%) were based on RNA variant data and about 12% were based on DNA variant data. The threefold gain of DNA-based NACs compared to the DNA fraction of all variants indicates an enormously increased likelihood for DNA variants to generally induce potential neoantigens. Although only a limited number of DNA-based NACs were detected, this observation may serve as a hint towards DNA as a more reliable data resource. Furthermore, this supports the necessity of vigilant multimodal validation when analyzing primary RNA data, as mentioned above. Different publications intensively discussed to which frequency somatic mutations generate potentially targetable neoantigens and consequently which fraction of tumors generate NACs that can elicit a sufficient immune response (Tran et al., 2015; Yarchoan et al., 2017; Parkhurst et al., 2019). In broad agreement, it was stated that only a tiny fraction of all somatic mutations give rise to cancer neoantigens that may potentially lead to spontaneous T cell responses (Lang et al., 2022). Numbers, however, vary depending on the entity, detection mode, and sensitivity. The analyzed data for variants and NACs of the ImmuNeo project suggest that based on the underlying pipeline, only a small number of candidates can be validated as immunogenic. However, this still does not show proof for being a functional cancer neoantigen.

With a fourfold increased NAC yield per RNA variant, patient 4 (renal cell carcinoma) was identified as a clear outlier with remarkable properties. A total of 14 NACs were generated by less than 3.500 RNA variants resulting in a yield of 0.4%. None of these peptides had favored restriction for HLA-A alleles (*Kd-method*). Most of them seemed to be restricted to HLA-C1203 (6 NACs) and HLA-B1501 (4 NACs). This is of particular interest since recent publications found evidence for a correlation between better clinical outcomes and a high number of HLA-A restricted neoepitopes in clear cell carcinomas (Matsushita et al., 2016).

It will be interesting to follow if the results of further analysis of HLA-B and HLA-C restricted neoepitope properties confirm these correlations found here.

4.4.2 Genetic origin

A major finding of the genetic variant type analysis that was realized with the ImmuNeo dataset indicates that the contribution of splice sites and intron variants to the identified NAC repertoire is much greater than a priori estimated by the fraction of splice sites

and intron variants across all DNA and RNA variants. As repeatedly stated by recent publications, increased alternative splicing events in tumors may be causal for a considerable repertoire of potentially immunogenic peptides (Xie et al., 2023). These may be induced by mutations in RNA cis-regulatory elements, trans-acting regulators, or the core spliceosome and may be up to 30% more abundant in tumors as in normal samples, as an analysis across 32 cancer types suggests, using The Cancer Genome Atlas (Kahles et al., 2018b).

4.4.3 Correlation between NAC source and binding affinity score

The detailed analysis of all NACs concerning their source revealed two major findings. First, those NACs that were based on variants identified on both DNA and RNA level exhibited, on average, a significantly higher binding affinity. Second, the same trend could be observed for NACs identified with both proteomic tools (pFind and PROSIT).

When considering the nano-molar binding affinity (Fig. 3.30(a)), seven out of eight NACs that were simultaneously detected on DNA and RNA level exhibit values below 100 nM. In comparison, for all NACs identified on RNA level only, a uniform spread over all affinity regimes could be observed (For DNA-only, the low number of NACs does not allow a conclusion). Despite this unambiguous result, it is striking that all of these RNA and DNA-based NACs show low to moderate variant frequencies. Together with the findings from the distributions of the tumor VF and total reads (Fig. 3.5 and 3.3), this may indicate a tendency of NACs with high tumor VF to result from variants with lower sequencing depths and hence higher risk of being false positive results. Alternatively these observations could be explained by immune editing events.

When considering the dependence of the nano-molar binding affinity on the used proteomic tool, a similar trend could be observed. An increased affinity with Kd values less than 100 nM could hence be observed for 19 out of 30 NACs that were covered by pFind and PROSIT at the same time. Here, NACs that were solely identified with one of both tools exhibited a significantly lower fraction of high-affinity candidates. (pFind: eight out of 50; PROSIT: five out of 13). Again, a slight tendency towards lower variant frequencies could be observed, however, with less explicit results. Besides the effect of being statistically more valid by the detection of two independent tools, this correlation could

also be explained through some entanglement between proteomic detection algorithms and binding predictors, both tending to prioritize sequences that, for some reasons, are "easier" to detect and hence have more available training data.

Both findings suggest that a combined approach, not only of exome and transcriptome data but also of different proteomic tools, could markedly improve response rates and function as a helpful selection method. On the one hand, this concept could be extended to **WGS!** to increase potential variant repertoires. On the other hand, the integration of further proteomic search algorithms, such as MSfragger (Kong et al., 2017), could substantially advance neoepitope discovery.

4.4.4 NAC-repertoire variability

Across the whole cohort, the patient-specific number of NACs was found to vary substantially. As indicated by further analysis of the corresponding immunopeptidomic data set by Tretter et al., the number of NACs strongly correlated with the size of the immunopeptidome. Hence, higher NAC-specificity due to matching MS-spectra to variants comes with the cost of reduced NAC-outcome since MS-approaches mapping only a fraction of the complete immunopeptidome of a tumor sample (Tretter et al., 2023). This limitation might be due to sample loss during the IP and the subsequent peptide purification processes (Zhang et al., 2019). Low instrument sensitivity, limitations of analysis strategies, and the requirement of abundant tissue samples are further shortcomings of MS/MS approaches (Macklin et al., 2020). Here it is suggested that optimizing machine-learning tools for the MS spectra matching and scoring procedures, together with increased sensitivity of MS instruments, will impact and improve MS discovery pipelines for neoantigens (Chong et al., 2022).

Important insight into the clinical impact is gained by combining different data sources. Correlating NAC yield and mutational load with clinical courses of the different entities is decisive for the revelation of potential associations and for linking immunological response behavior to directly measurable and quantifiable parameters.

4.5 Experimental validation with acDC assay

4.5.1 Advantages of dendritic co-culture

There are different experimental methods for the detection of specific T-cell responses. Whereas long-known approaches using multimerized pMHC ligands for a given population of T cells require a priori knowledge of the MHC restriction of each peptide, the assay used in this work was shown to sensitively detect T cell immune responses also with scarce patient material (Martinuzzi et al., 2011; Hadrup et al., 2009). Since no separate generation of antigen-presenting cells was possible, cell co-cultivation in combination with a highly sensitive ELISpot measurement provided the most feasible option.

Though experimental procedures are tedious, the realized approach offers the possibility to directly assess T cell functionality upon specific stimulation in contrast to mere antigen binding capacity, which is analyzed in pMHC approaches. Besides, the quality of the assay outcome can be monitored by observation of T cell growth. Yet, signs of T cell exhaustion may hamper the detectability of present neoantigen-specific T cell responses and the aspect of only one assessed cytokine IFN-γ places a crucial limitation for the detection of immune responses in general. Furthermore the application of fixed concentrations of peptides for the stimulation might limit the scope of detection. Besides, the specific controls facilitate a good signal-to-noise ratio, especially for higher cell numbers.

4.5.2 Quality of primary patient samples

All realized assays that intended to confirm T cell immunogenicity against the identified epitopes were done with freshly thawed PBMCs. However, there were huge differences between the different patients concerning the sample quality. By the time of thawing, samples from some patients exhibited strong tendencies to clot together, forming a sticky cell bulk which was connected with lower cell number outcome and mostly lower cell viability. For future assays, it seems indispensable to precisely survey this phenomenon, which was already described in earlier publications, and eventually avoid a loss of cells by incorporation of a DNAse (benzonase nuclease) treatment (García-Piñeres et al., 2006; Smith et al., 2001).

The general problem of overall high cell number variations and low cell yield for particular patients was responsible for certain limitations that resulted in a higher likelihood of missing relevant immune responses. In order to distinguish between the required conditions pulsed, unpulsed, and irrelevantly pulsed, the number of cells was only sufficient for singlet approaches on assay Day 1 (exception: experiment *acDC 01# 4* with Triplets) and duplicate approaches on assay Day 13. Realizing triplicates of each condition would probably have reduced statistical aberrations and hence was implemented into the ongoing projects later if possible.

4.5.3 Technical considerations

The readout procedure, which was done with the ELISpot analyzer *ImmunoSpot S6 Ultra-V* required different threshold settings, which were adjusted according to the quality of the scanned image of the ELISpot plate, but also on the individual properties of the formed colonies. As a result, some of the analysis had to be done with different threshold settings, which in principle, could have affected the comparability of the results. However, this error can be neglected due to generally low colony counts. The image quality, directly linked to the focus level of acquired pictures, was another source of variabilities. Low sharpness of the acquired images might have significantly affected counting events of nearby SFUs, which, however, was individually controlled after readout and, in the case of our results, a rarely seen event. Of note, due to excessive dot density, the positive control was not considered the same way but only served as a qualitative control. Guidelines for standardized ELISpot evaluation by Janetzki et al. may help to adjust corresponding reading parameters to increase the reliability and comparability of the results (Janetzki et al., 2015). Other important consensus initiatives that are important for good manufacturing practice (GMP) and clinical translation could only partially be complied with due to limited patient material (Moodie et al., 2010).

4.6 Conceptional considerations of multi-omics approaches

The ImmuNeo discovery pipeline for tumor neoantigens, whose approaches and results are described in this thesis, is based on several individually developed sub-projects re-

alized by different research groups. The combined workflow aiming to merge these groups' results comes with different challenges and structural limitations that will be discussed in the following. It is important to mention that all described aspects are especially relevant for further clinical translation and exceed the feasibility and the financial means of this project.

Standardized sample acquisition and preparation were required since all subsequent considerations are based on the primary patient sample. However, different tumor entities are clinically treated by different surgical units and hence different specialists, each time with its own methodology and specifics. Furthermore, tumor sample withdrawal was realized over various institutions and sites with individual standard operating procedures (SOPs). The observed variation of tumor sample sizes and qualities, which for the most part may be a result of individual tumor status and hence of its evolution and growths, complicated subsequent sample preparation standards and led to different prerequisites for further analysis like **WES!** or MS/MS. The macroscopic texture (soft vs. solid) and the fraction of potentially necrotic tumor parts or hypoxic areas are additional unswayable parameters that must be mentioned in this context. This challenge became especially explicit for RNA-sequencing analysis, where lacking sample material led to missing RNA data in eight cases.

Intra-tumor heterogeneity constitutes another closely linked challenge not only for comprehensive data acquisition. Earlier studies analyzing intratumor heterogeneity in primary renal carcinoma claimed that more than 63% of all somatic mutations were not detectable across every tumor region (Gerlinger et al., 2012). This suggests that, i.e., incomplete resections or not incomprehensively prepared samples might not reflect the complete mutational landscape of a tumor, which is essential for the detection of subclonal tumor cell populations. In recent publications, it was even shown that single-sample reconstructions of subclonal populations systematically underestimate intra-tumoural heterogeneity, indicating that interpretations of specific architectures and subclonal variants should be made cautiously (Liu et al., 2020). In a clinical context, intra-tumor heterogeneity poses widely known therapeutic limitations since heterogeneity might not only be required for but can even promote tumor development and progression as suggested earlier (McGranahan & Swanton, 2015). As clonal diversity (the size of sub-clonal fractions) has shown to be associated with poor clinical outcome after treatment with chemo-

and radiotherapy (Andor et al., 2016), it might also impose certain limitations for tumor neoantigen related therapeutic approaches (El-Sayes et al., 2021; Xie et al., 2022; Sanli et al., 2019). This becomes evident as different strategies of immune evasion in terms of quantitative modulation or qualitative alteration of the presented antigen repertoires, such as modulated antigen expression levels or HLA-I surface levels, have been observed in the past (Jhunjhunwala et al., 2021). Besides, since tumor evolution underlies not only genetic mutations but also epigenetic and transcriptomic alterations, newer studies claim that exclusively genomic approaches fail to explain the evolution of ITH (Black & McGranahan, 2021), hence necessitating multi-omics approaches to characterize tumor mutanomes comprehensively.

To face this limitation of exclusive genomic analyses, RNA-seq data was included and formed an integral part of the ImmuNeo pipeline. However, data artifacts were more likely than for DNA data. Here a missing specific control for RNA variants that derive from RNA processing events might be causal for the contingently high abundance of artifacts in RNA data. No control with healthy (normal) tissue could be done here since these variants cannot be validated by matched-normal DNA samples (cp. paragraph on implementation of GTex in Section 4.2.2).

A pipeline-specific but thus structural limitation was caused by the interdependence of different sub-pipelines whose outcomes were crucial for further analysis and pipeline development. Due to the nature of combining several on-the-edge technologies, an inevitable need for continuous adaption of different methods to novel insights in each research field necessitated continuous adaption of each sub-project. Here small pipeline-related changes, i.e., updates in bioinformatical databases or pipeline improvements, caused tremendous changes in the approval of possible NACs, which could then completely alter priory established prioritization strategies, and results. Furthermore, experimentally confirmed candidates had to be reconsidered and re-evaluated distinctly. These technical issues indicate that the bioinformatical procedures concerning Genomics, Transcriptomics, and Proteomics data are still shaped by ongoing development. Continuous improvements are warranted and ongoing to obtain robust pipelines for further analysis.

List of Abbreviations

AA amino acid

acDC accelerated co-cultured dendritic cell

ACT adoptive cell therapy

ADC antibody-drug conjugate

ADCC antibody dependent cellular cytotoxicity

ADCP antibody-dependent cellular phagocytosis

AF allele frequency

ALT altered nucleotide

ANN artificial neural network

APC antigen-presenting cell

APP antigen-processing and presentation

B-ALL B cell acute lymphoblastic leukaemia

BCMA B cell maturation antigen

BLAST basic local alignment search tool

CAR chimeric antigen receptor

CAR chimeric antigen receptor

CDC complement-dependent cytotoxicity

CHROM chromosome number

cRPMI complete RPMI

CTA cancer testis antigen

CTLA4 cytotoxic T-lymphocyte-associated antigen 4

dbSNP SNP database

DRiPs defective ribosomal products

EBV Epstein-Barr virus

EGFR epidermal Growth Factor Receptor

EGFR epidermal growth factor receptor

EL eluted ligand

FDR false discovery rate

GMP good manufacturing practice

GM-CSF granulocyte-macrophage colony-stimulating factor

GTex Genotype-Tissue Expression

HBV hepatitis B virus

HER2 human epidermal growth factor receptor 2

HLA human leukocyte antigen

HMM hidden Markov model

HPV human papillomavirus

ICI immune checkpoint inhibition

IEDB immune epitope database

IFN-γ interferon-γ

IL-1β interleukin-1β

IL-4 interleukin 4

IL-7 interleukin 7

indel insertion-deletion mutation

irAE immune-related adverse event

ITH intratumor heterogeneity

LAG-3 lymphocyte-activated gene 3

LCL lymphoblastoid cell line

mAb monoclonal antibody

MAP MHC class I–associated peptide

MHC major histocompatibility complex

MMRD mismatch repair deficiency

MPNST malignant peripheral nerve sheath tumor

MS mass spectrometry

MS/MS tandem mass spectrometry

NA neoantigen

NAC proteogenomic neoantigen candidate

NCBI National Center for Biotechnology Information

NGS next-generation sequencing

NSCLC non-small-cell lung cancer

NSM non-synonymous mutation

nuORF novel or unannotated open reading frame

ORF open reading frame

PBMC peripheral blood mononuclear cell

PD-1 programmed cell death 1 receptor

PD-L1 programmed cell death receptor ligand 1

pHLA HLA class I peptide

pip Python package installer

PMA tetradecanoyl phorbol acetate

pMHC peptide MHC

POS position of the base pair

PSP proteasome-spliced peptide

QSAR quantitative structure-affinity relationship

REF reference nucleotide

RNA-Seq RNA sequencing

RNA ribonucleic acid

SFU spot forming unit

SNP single nucleotide polymorphism

SNV single nucleotide variant

SOP standard operating procedure

Strep-HRP streptavidin horseradish peroxidase

TAA tumor associated antigen

TAP transporter associated with antigen processing

TIGIT T cell immunoreceptor with Ig and ITIM domain

TIL tumor-infiltrating lymphocyte

TMB tumor mutational burden

TME tumor microenvironment

TNF-α tumor necrosis factor-α

TSA tumor specific antigen

UCSC University of California Santa Cruz

VCF variant call format

VEGF vascular endothelial growth factor

VF variant frequency

VISTA V-domain Ig suppressor of T cell activation

WES whole exome sequencing

WGS whole genome sequencing

WT-DNA wild type DNA

WT wildtype

List of Figures

List of Tables

References

Abelin, J. G., Keskin, D. B., Sarkizova, S., Hartigan, C. R., Zhang, W., Sidney, J., ... Wu, C. J. (2017). Mass spectrometry profiling of hla-associated peptidomes in mono-allelic cells enables more accurate epitope prediction. *Immunity*, *46*(2), 315–326. doi: 10.1016/j.immuni.2017.02.007.

Abualrous, E. T., Sticht, J., and Freund, C. (2021). Major histocompatibility complex (mhc) class i and class ii proteins: impact of polymorphism on antigen presentation. *Current Opinion in Immunology*, *70*, 95–104. doi: 10.1016/j.coi.2021.04.009.

Akinleye, A. and Rasool, Z. (2019). Immune checkpoint inhibitors of pd-l1 as cancer therapeutics. *Journal of Hematology & Oncology*, *12*(1), 92. doi: 10.1186/s13045-019-0779-5.

Alexandrov, L. B., Nik-Zainal, S., Wedge, D. C., Aparicio, S. A. J. R., Behjati, S., Biankin, A. V., ... Stratton, M. R. (2013a). Signatures of mutational processes in human cancer. *Nature*, *500*(74637463), 415–421. doi: 10.1038/nature12477.

Alexandrov, L. B., Nik-Zainal, S., Wedge, D. C., Aparicio, S. A. J. R., Behjati, S., Biankin, A. V., ... Stratton, M. R. (2013b). Signatures of mutational processes in human cancer. *Nature*, *500*(74637463), 415–421. doi: 10.1038/nature12477.

Alspach, E., Lussier, D. M., Miceli, A. P., Kizhvatov, I., DuPage, M., Luoma, A. M., ... Schreiber, R. D. (2019). Mhc-ii neoantigens shape tumour immunity and response to immunotherapy. *Nature*, *574*(77807780), 696–701. doi: 10.1038/s41586-019-1671-8.

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

Anderson, M. A. and Gusella, J. F. (1984). Use of cyclosporin a in establishing epstein-

barr virus-transformed human lymphoblastoid cell lines. *In Vitro*, *20*(11), 856–858. doi: 10.1007/BF02619631.
URL http://link.springer.com/10.1007/BF02619631

Andor, N., Graham, T. A., Jansen, M., Xia, L. C., Aktipis, C. A., Petritsch, C., ... Maley, C. C. (2016). Pan-cancer analysis of the extent and consequences of intratumor heterogeneity. *Nature Medicine*, *22*(11), 105–113. doi: 10.1038/nm.3984.

Andreatta, M. and Nielsen, M. (2016). Gapped sequence alignment using artificial neural networks: application to the MHC class I system. *Bioinformatics*, *32*(4), 511–517. doi: 10.1093/bioinformatics/btv639.
URL https://academic.oup.com/bioinformatics/article/32/4/511/1744469

Andrews, L. P., Marciscano, A. E., Drake, C. G., and Vignali, D. A. A. (2017). Lag3 (cd223) as a cancer immunotherapy target. *Immunological Reviews*, *276*(1), 80–96. doi: 10.1111/imr.12519.

Ansell, S. M., Radford, J., Connors, J. M., Długosz-Danecka, M., Kim, W.-S., Gallamini, A., ... Straus, D. J. (2022). Overall survival with brentuximab vedotin in stage iii or iv hodgkin's lymphoma. *New England Journal of Medicine*, *387*(4), 310–320. doi: 10.1056/NEJMoa2206125.

Antonia, S. J., Villegas, A., Daniel, D., Vicente, D., Murakami, S., Hui, R., ... Özgüroğlu, M. (2018). Overall survival with durvalumab after chemoradiotherapy in stage iii nsclc. *New England Journal of Medicine*, *379*(24), 2342–2350. doi: 10.1056/NEJMoa1809697.

Arnaud, M., Duchamp, M., Bobisse, S., Renaud, P., Coukos, G., and Harari, A. (2020). Biotechnologies to tackle the challenge of neoantigen identification. *Current Opinion in Biotechnology*, *65*, 52–59. doi: 10.1016/j.copbio.2019.12.014.

Arzumanyan, A., Reis, H. M. G. P. V., and Feitelson, M. A. (2013). Pathogenic mechanisms in hbv- and hcv-associated hepatocellular carcinoma. *Nature Reviews Cancer*, *13*(22), 123–135. doi: 10.1038/nrc3449.

Bassani-Sternberg, M. (2018). *Mass Spectrometry Based Immunopeptidomics for the Discovery of Cancer Neoantigens*, (p. 209–221). Methods in Molecular Biology. New York, NY:

Springer. doi: 10.1007/978-1-4939-7537-2_14.
URL https://doi.org/10.1007/978-1-4939-7537-2_14

Bassani-Sternberg, M., Bräunlein, E., Klar, R., Engleitner, T., Sinitcyn, P., Audehm, S., ... Krackhardt, A. M. (2016). Direct identification of clinically relevant neoepitopes presented on native human melanoma tissue by mass spectrometry. *Nature Communications*, *7*(11), 13404. doi: 10.1038/ncomms13404.

Bassani-Sternberg, M., Pletscher-Frankild, S., Jensen, L. J., and Mann, M. (2015). Mass spectrometry of human leukocyte antigen class i peptidomes reveals strong effects of protein abundance and turnover on antigen presentation *[s]. *Molecular & Cellular Proteomics*, *14*(3), 658–673. doi: 10.1074/mcp.M114.042812.

Benjamin, D., Sato, T., Cibulskis, K., Getz, G., Stewart, C., and Lichtenstein, L. (2019). Calling somatic snvs and indels with mutect2. *BioRxiv*, (p. 861054).

Benmebarek, M.-R., Karches, C. H., Cadilha, B. L., Lesch, S., Endres, S., and Kobold, S. (2019). Killing mechanisms of chimeric antigen receptor (car) t cells. *International Journal of Molecular Sciences*, *20*(66), 1283. doi: 10.3390/ijms20061283.

Bhojwani, D., Sposto, R., Shah, N. N., Rodriguez, V., Yuan, C., Stetler-Stevenson, M., ... Rheingold, S. R. (2019). Inotuzumab ozogamicin in pediatric patients with relapsed/refractory acute lymphoblastic leukemia. *Leukemia*, *33*(44), 884–892. doi: 10.1038/s41375-018-0265-z.

Bjerregaard, A.-M., Nielsen, M., Jurtz, V., Barra, C. M., Hadrup, S. R., Szallasi, Z. and Eklund, A. C. (2017). An analysis of natural t cell responses to predicted tumor neoepitopes. *Frontiers in Immunology*, *8*.
URL https://www.frontiersin.org/articles/10.3389/fimmu.2017.01566

Black, J. R. M. and McGranahan, N. (2021). Genetic and non-genetic clonal diversity in cancer evolution. *Nature Reviews Cancer*, *21*(66), 379–392. doi: 10.1038/s41568-021-00336-2.

Bonanni, P. (1999). Demographic impact of vaccination: a review. *Vaccine*, *17*, S120–S125. doi: 10.1016/S0264-410X(99)00306-0.

Bonifant, C. L., Jackson, H. J., Brentjens, R. J., and Curran, K. J. (2016). Toxicity and management in car t-cell therapy. *Molecular Therapy - Oncolytics*, *3*, 16011. doi: 10.1038/mto.2016.11.

Bonsack, M., Hoppe, S., Winter, J., Tichy, D., Zeller, C., Küpper, M. D., ... Riemer, A. B. (2019). Performance evaluation of mhc class-i binding prediction tools based on an experimentally validated mhc–peptide binding data set. *Cancer Immunology Research*, *7*(5), 719–736. doi: 10.1158/2326-6066.CIR-18-0584.

Boratyn, G. M., Camacho, C., Cooper, P. S., Coulouris, G., Fong, A., Ma, N., ... Zaretskaya, I. (2013). BLAST: a more efficient report with usability improvements. *Nucleic Acids Research*, *41*(W1), W29–W33. doi: 10.1093/nar/gkt282.
URL http://academic.oup.com/nar/article/41/W1/W29/1091045/BLAST-a-more-efficient-report-with-usability

Borghaei, H., Paz-Ares, L., Horn, L., Spigel, D. R., Steins, M., Ready, N. E., ... Brahmer, J. R. (2015). Nivolumab versus docetaxel in advanced nonsquamous non–small-cell lung cancer. *New England Journal of Medicine*, *373*(17), 1627–1639. doi: 10.1056/NEJMoa1507643.

Brahmer, J. R., Lacchetti, C., Schneider, B. J., Atkins, M. B., Brassil, K. J., Caterino, J. M., ... Thompson, J. A. (2018). Management of immune-related adverse events in patients treated with immune checkpoint inhibitor therapy: American society of clinical oncology clinical practice guideline. *Journal of Clinical Oncology*, *36*(17), 1714–1768. doi: 10.1200/JCO.2017.77.6385.

Brudno, J. N. and Kochenderfer, J. N. (2018). Chimeric antigen receptor t-cell therapies for lymphoma. *Nature Reviews Clinical Oncology*, *15*(11), 31–46. doi: 10.1038/nrclinonc.2017.128.

Bui, J. D. and Schreiber, R. D. (2007). Cancer immunosurveillance, immunoediting and inflammation: independent or interdependent processes? *Current Opinion in Immunology*, *19*(2), 203–208. doi: 10.1016/j.coi.2007.02.001.

Bulik-Sullivan, B., Busby, J., Palmer, C. D., Davis, M. J., Murphy, T., Clark, A., ... Yelensky, R. (2019). Deep learning using tumor hla peptide mass spectrometry datasets improves

neoantigen identification. *Nature Biotechnology*, *37*(11), 55–63. doi: 10.1038/nbt.4313.

Burke, K. P., Grebinoski, S., Sharpe, A. H., and Vignali, D. A. (2020). Understanding adverse events of immunotherapy: A mechanistic perspective. *Journal of Experimental Medicine*, *218*(1), e20192179. doi: 10.1084/jem.20192179.

Caron, E., Kowalewski, D., Koh, C. C., Sturm, T., Schuster, H., and Aebersold, R. (2015). Analysis of major histocompatibility complex (mhc) immunopeptidomes using mass spectrometry*. *Molecular & Cellular Proteomics*, *14*(12), 3105–3117. doi: 10.1074/mcp.O115.052431.

Castle, J. C., Kreiter, S., Diekmann, J., Löwer, M., van de Roemer, N., de Graaf, J., ... Sahin, U. (2012). Exploiting the mutanome for tumor vaccination. *Cancer Research*, *72*(5), 1081–1091. doi: 10.1158/0008-5472.CAN-11-3722.

Chalmers, Z. R., Connelly, C. F., Fabrizio, D., Gay, L., Ali, S. M., Ennis, R., ... Frampton, G. M. (2017). Analysis of 100,000 human cancer genomes reveals the landscape of tumor mutational burden. *Genome Medicine*, *9*(1), 34. doi: 10.1186/s13073-017-0424-2.

Chan, T. A., Yarchoan, M., Jaffee, E., Swanton, C., Quezada, S. A., Stenzinger, A. and Peters, S. (2019). Development of tumor mutation burden as an immunotherapy biomarker: utility for the oncology clinic. *Annals of Oncology*, *30*(1), 44–56. doi: 10.1093/annonc/mdy495.

Chau, C. H., Steeg, P. S., and Figg, W. D. (2019). Antibody–drug conjugates for cancer. *The Lancet*, *394*(10200), 793–804. doi: 10.1016/S0140-6736(19)31774-X.

Chauvin, J.-M. and Zarour, H. M. (2020). Tigit in cancer immunotherapy. *Journal for Immunotherapy of Cancer*, *8*(2), e000957. doi: 10.1136/jitc-2020-000957.

Chen, J.-S., Lan, K., and Hung, M.-C. (2003). Strategies to target her2/neu overexpression for cancer therapy. *Drug Resistance Updates*, *6*(3), 129–136. doi: 10.1016/S1368-7646(03)00040-2.

Chi, H., He, K., Yang, B., Chen, Z., Sun, R.-X., Fan, S.-B., ... He, S.-M. (2015). pfind-alioth: A novel unrestricted database search algorithm to improve the interpretation of high-resolution ms/ms data. *Journal of proteomics*, *125*, 89—97. doi: 10.1016/j.jprot.2015.05.

009.
URL https://doi.org/10.1016/j.jprot.2015.05.009

Chi, H., Liu, C., Yang, H., Zeng, W.-F., Wu, L., Zhou, W.-J., ... He, S.-M. (2018). OpenpFind enables precise, comprehensive and rapid peptide identification in shotgun proteomics. preprint, Bioinformatics.
URL http://biorxiv.org/lookup/doi/10.1101/285395

Chocarro, L., Blanco, E., Arasanz, H., Fernández-Rubio, L., Bocanegra, A., Echaide, M., ... Escors, D. (2022). Clinical landscape of lag-3-targeted therapy. *Immuno-Oncology and Technology*, *14*, 100079. doi: 10.1016/j.iotech.2022.100079.

Chong, C., Coukos, G., and Bassani-Sternberg, M. (2022). Identification of tumor antigens with immunopeptidomics. *Nature Biotechnology*, *40*(22), 175–188. doi: 10.1038/s41587-021-01038-8.

Conev, A., Devaurs, D., Rigo, M. M., Antunes, D. A., and Kavraki, L. E. (2022). 3phla-score improves structure-based peptide-hla binding affinity prediction. *Scientific Reports*, *12*(11), 10749. doi: 10.1038/s41598-022-14526-x.

Cox, J. and 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. doi: 10.1038/nbt.1511.
URL http://www.nature.com/articles/nbt.1511

Dersh, D., Hollý, J., and Yewdell, J. W. (2021). A few good peptides: Mhc class i-based cancer immunosurveillance and immunoevasion. *Nature Reviews Immunology*, *21*(22), 116–128. doi: 10.1038/s41577-020-0390-6.

Dhatchinamoorthy, K., Colbert, J. D., and Rock, K. L. (2021). Cancer immune evasion through loss of mhc class i antigen presentation. *Frontiers in Immunology*, *12*.
URL https://www.frontiersin.org/articles/10.3389/fimmu.2021.636568

Drago, J. Z., Modi, S., and Chandarlapaty, S. (2021). Unlocking the potential of antibody–drug conjugates for cancer therapy. *Nature Reviews Clinical Oncology*, *18*(66), 327–344. doi: 10.1038/s41571-021-00470-8.

D'Angelo, S. P., Russell, J., Lebbé, C., Chmielowski, B., Gambichler, T., Grob, J.-J., ... Kaufman, H. L. (2018). Efficacy and safety of first-line avelumab treatment in patients with stage iv metastatic merkel cell carcinoma: A preplanned interim analysis of a clinical trial. *JAMA Oncology*, *4*(9), e180077. doi: 10.1001/jamaoncol.2018.0077.

Eisenberg, E. and Levanon, E. Y. (2018). A-to-i rna editing — immune protector and transcriptome diversifier. *Nature Reviews Genetics*, *19*(88), 473–490. doi: 10.1038/s41576-018-0006-1.

El-Sayes, N., Vito, A., and Mossman, K. (2021). Tumor heterogeneity: A great barrier in the age of cancer immunotherapy. *Cancers*, *13*(44), 806. doi: 10.3390/cancers13040806.

Erhard, F., Halenius, A., Zimmermann, C., L'Hernault, A., Kowalewski, D. J., Weekes, M. P., ... Dölken, L. (2018). Improved ribo-seq enables identification of cryptic translation events. *Nature Methods*, *15*(55), 363–366. doi: 10.1038/nmeth.4631.

Esfahani, K., Roudaia, L., Buhlaiga, N., Del Rincon, S. V., Papneja, N., and Miller, W. H. (2020). A review of cancer immunotherapy: From the past, to the present, to the future. *Current Oncology*, *27*(s2s2), 87–97. doi: 10.3747/co.27.5223.

Falcaro, M., Castañon, A., Ndlela, B., Checchi, M., Soldan, K., Lopez-Bernal, J., ... Sasieni, P. (2021). The effects of the national hpv vaccination programme in england, uk, on cervical cancer and grade 3 cervical intraepithelial neoplasia incidence: a register-based observational study. *The Lancet*, *398*(10316), 2084–2092. doi: 10.1016/S0140-6736(21)02178-4.

Farkona, S., Diamandis, E. P., and Blasutig, I. M. (2016). Cancer immunotherapy: the beginning of the end of cancer? *BMC Medicine*, *14*(1), 73. doi: 10.1186/s12916-016-0623-5.

Garcia-Garijo, A., Fajardo, C. A., and Gros, A. (2019). Determinants for neoantigen identification. *Frontiers in Immunology*, *10*.
URL `https://www.frontiersin.org/articles/10.3389/fimmu.2019.01392`

García-Piñeres, A. J., Hildesheim, A., Williams, M., Trivett, M., Strobl, S., and Pinto, L. A. (2006). Dnase treatment following thawing of cryopreserved pbmc is a procedure suitable for lymphocyte functional studies. *Journal of Immunological Methods*, *313*(1),

209–213. doi: 10.1016/j.jim.2006.04.004.

Garon, E. B., Rizvi, N. A., Hui, R., Leighl, N., Balmanoukian, A. S., Eder, J. P., ... Gandhi, L. (2015). Pembrolizumab for the treatment of non–small-cell lung cancer. *New England Journal of Medicine*, *372*(21), 2018–2028. doi: 10.1056/NEJMoa1501824.

Gerlinger, M., Rowan, A. J., Horswell, S., Larkin, J., Endesfelder, D., Gronroos, E., ... Swanton, C. (2012). Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *New England Journal of Medicine*, *366*(10), 883–892. doi: 10.1056/NEJMoa1113205.

Gessulat, S., Schmidt, T., Zolg, D. P., Samaras, P., Schnatbaum, K., Zerweck, J., ... Wilhelm, M. (2019). Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods*, *16*(6), 509–518. doi: 10.1038/s41592-019-0426-7.
URL http://www.nature.com/articles/s41592-019-0426-7

Gfeller, D. and Bassani-Sternberg, M. (2018). Predicting Antigen Presentation—What Could We Learn From a Million Peptides? *Frontiers in Immunology*, *9*, 1716. doi: 10.3389/fimmu.2018.01716.
URL https://www.frontiersin.org/article/10.3389/fimmu.2018.01716/full

Gonzalez-Galarza, F. F., McCabe, A., Santos, E. J. M. d., Jones, J., Takeshita, L., Ortega-Rivera, N. D., ... Jones, A. R. (2019). 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. doi: 10.1093/nar/gkz1029.
URL https://doi.org/10.1093/nar/gkz1029

Grant, R. C., Denroche, R., Jang, G. H., Nowak, K. M., Zhang, A., Borgida, A., ... Gallinger, S. (2021). Clinical and genomic characterisation of mismatch repair deficient pancreatic adenocarcinoma. *Gut*, *70*(10), 1894–1903. doi: 10.1136/gutjnl-2020-320730.

Griffith, M., Miller, C. A., Griffith, O. L., Krysiak, K., Skidmore, Z. L., Ramu, A., ... Wilson, R. K. (2015). Optimizing cancer genome sequencing and analysis. *Cell systems*, *1*(3), 210–223. doi: 10.1016/j.cels.2015.08.015.

Gubin, M. M., Artyomov, M. N., Mardis, E. R., and Schreiber, R. D. (2015). Tumor neoanti-

gens: building a framework for personalized cancer immunotherapy. *The Journal of Clinical Investigation*, *125*(9), 3413–3421. doi: 10.1172/JCI80008.

Gui, P. and Bivona, T. G. (2022). Evolution of metastasis: new tools and insights. *Trends in Cancer*, *8*(2), 98–109. doi: 10.1016/j.trecan.2021.11.002.

Hadrup, S. R., Bakker, A. H., Shu, C. J., Andersen, R. S., van Veluw, J., Hombrink, P., ... Schumacher, T. N. (2009). Parallel detection of antigen-specific t-cell responses by multidimensional encoding of mhc multimers. *Nature Methods*, *6*(77), 520–526. doi: 10.1038/nmeth.1345.

Hanna, J., M. G. and Peters, L. C. (1978). Immunotherapy of established micrometastases with bacillus calmette-guérin tumor cell vaccine1. *Cancer Research*, *38*(1), 204–209.

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

Heemskerk, B., Kvistborg, P., and Schumacher, T. N. M. (2013). The cancer antigenome. *The EMBO Journal*, *32*(2), 194–203. doi: 10.1038/emboj.2012.333.

Hegde, P. S. and Chen, D. S. (2020). Top 10 challenges in cancer immunotherapy. *Immunity*, *52*(1), 17–35. doi: 10.1016/j.immuni.2019.12.011.

Hodi, F. S., O'Day, S. J., McDermott, D. F., Weber, R. W., Sosman, J. A., Haanen, J. B., ... Urba, W. J. (2010). Improved survival with ipilimumab in patients with metastatic melanoma. *New England Journal of Medicine*, *363*(8), 711–723. doi: 10.1056/NEJMoa1003466.

Hogquist, K. A., Baldwin, T. A., and Jameson, S. C. (2005). Central tolerance: learning self-control in the thymus. *Nature Reviews Immunology*, *5*(1010), 772–782. doi: 10.1038/nri1707.

Horak, P., Heining, C., Kreutzfeldt, S., Hutter, B., Mock, A., Hüllein, J. ... (2021). Comprehensive genomic and transcriptomic analysis for guiding therapeutic decisions in patients with rare cancers. *Cancer discovery*, *11*(11), 2780–2795.

Hoyos, L. E. and Abdel-Wahab, O. (2018). Cancer-specific splicing changes and the po-

tential for splicing-derived neoantigens. *Cancer Cell*, *34*(2), 181–183. doi: 10.1016/j.ccell.2018.07.008.

Hu, Z., Li, Z., Ma, Z., and Curtis, C. (2020). Multi-cancer analysis of clonality and the timing of systemic spread in paired primary tumors and metastases. *Nature Genetics*, *52*(77), 701–708. doi: 10.1038/s41588-020-0628-z.

Inman, B. A., Longo, T. A., Ramalingam, S., and Harrison, M. R. (2017). Atezolizumab: A pd-l1–blocking antibody for bladder cancer. *Clinical Cancer Research*, *23*(8), 1886–1890. doi: 10.1158/1078-0432.CCR-16-1417.

Jacoby, M. A., Duncavage, E. J., and Walter, M. J. (2015). Implications of tumor clonal heterogeneity in the era of next-generation sequencing. *Trends in Cancer*, *1*(4), 231–241. doi: 10.1016/j.trecan.2015.10.006.

Janetzki, S., Price, L., Schroeder, H., Britten, C. M., Welters, M. J. P., and Hoos, A. (2015). Guidelines for the automated evaluation of elispot assays. *Nature Protocols*, *10*(77), 1098–1115. doi: 10.1038/nprot.2015.068.

Jhunjhunwala, S., Hammer, C., and Delamarre, L. (2021). Antigen presentation in cancer: insights into tumour immunogenicity and immune evasion. *Nature Reviews Cancer*, *21*(55), 298–312. doi: 10.1038/s41568-021-00339-z.

Jiang, T., Shi, T., Zhang, H., Hu, J., Song, Y., Wei, J., ... Zhou, C. (2019). Tumor neoantigens: from basic research to clinical applications. *Journal of Hematology & Oncology*, *12*(1), 93. doi: 10.1186/s13045-019-0787-5.

Johnson, M., Zaretskaya, I., Raytselis, Y., Merezhuk, Y., McGinnis, S., and Madden, T. L. (2008). NCBI BLAST: a better web interface. *Nucleic Acids Research*, *36*(Web Server), W5–W9. doi: 10.1093/nar/gkn201.
URL `https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkn201`

Jones, S., Chen, W.-d., Parmigiani, G., Diehl, F., Beerenwinkel, N., Antal, T., ... Markowitz, S. D. (2008). Comparative lesion sequencing provides insights into tumor evolution. *Proceedings of the National Academy of Sciences*, *105*(11), 4283–4288. doi: 10.1073/pnas.0712345105.

Jurtz, V., Paul, S., Andreatta, M., Marcatili, P., Peters, B., and Nielsen, M. (2017). Netmhcpan-4.0: Improved peptide–mhc class i interaction predictions integrating eluted ligand and peptide binding affinity data. *The Journal of Immunology*, *199*(9), 3360–3368. doi: 10.4049/jimmunol.1700893.

Kahles, A., Lehmann, K.-V., Toussaint, N. C., Hüser, M., Stark, S. G., Sachsenberg, T., ... Rätsch, G. (2018a). Comprehensive analysis of alternative splicing across tumors from 8,705 patients. *Cancer Cell*, *34*(2), 211–224.e6. doi: 10.1016/j.ccell.2018.07.001.

Kahles, A., Lehmann, K.-V., Toussaint, N. C., Hüser, M., Stark, S. G., Sachsenberg, T., ... Rätsch, G. (2018b). Comprehensive analysis of alternative splicing across tumors from 8,705 patients. *Cancer Cell*, *34*(2), 211–224.e6. doi: 10.1016/j.ccell.2018.07.001.

Kent, W. J. (2002). Blat—the blast-like alignment tool. *Genome Research*, *12*(4), 656–664. doi: 10.1101/gr.229202, company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press publisher: Cold Spring Harbor Lab PMID: 11932250.

Kim, S., Scheffler, K., Halpern, A. L., Bekritsky, M. A., Noh, E., Källberg, M. ... (2018). Strelka2: fast and accurate calling of germline and somatic variants. *Nature methods*, *15*(8), 591–594.

Kimiz-Gebologlu, I., Gulce-Iz, S., and Biray-Avci, C. (2018). Monoclonal antibodies in cancer immunotherapy. *Molecular Biology Reports*, *45*(6), 2935–2940. doi: 10.1007/s11033-018-4427-x.

Kong, A. T., Leprevost, F. V., Avtonomov, D. M., Mellacheruvu, D., and Nesvizhskii, A. I. (2017). Msfragger: ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nature Methods*, *14*(55), 513–520. doi: 10.1038/nmeth.4256.

Kuhn, R. M., Haussler, D., and Kent, W. J. (2013). The ucsc genome browser and associated tools. *Briefings in Bioinformatics*, *14*(2), 144–161. doi: 10.1093/bib/bbs038.

Lang, F., Schrörs, B., Löwer, M., Türeci, O., and Sahin, U. (2022). Identification of neoantigens for individualized therapeutic cancer vaccines. *Nature Reviews Drug Discovery*,

21(44), 261–282. doi: 10.1038/s41573-021-00387-y.

Lange, S., Engleitner, T., Mueller, S., Maresch, R., Zwiebel, M., Gonzalez-Silva, L. ... (2020). Analysis pipelines for cancer genome sequencing in mice. *Nature Protocols*, *15*(2), 266–315.

Larsen, M. V., Lundegaard, C., Lamberth, K., Buus, S., Brunak, S., Lund, O. and Nielsen, M. (2005). An integrative approach to ctl epitope prediction: A combined algorithm integrating mhc class i binding, tap transport efficiency, and proteasomal cleavage predictions. *European Journal of Immunology*, *35*(8), 2295–2303. doi: 10.1002/eji.200425811.

Laumont, C. M., Daouda, T., Laverdure, J.-P., Bonneil, E., Caron-Lizotte, O., Hardy, M.-P., ... Perreault, C. (2016). Global proteogenomic analysis of human mhc class i-associated peptides derived from non-canonical reading frames. *Nature Communications*, *7*(11), 10238. doi: 10.1038/ncomms10238.

Laumont, C. M., Vincent, K., Hesnard, L., Audemard, E., Bonneil, E., Laverdure, J.-P., ... Perreault, C. (2018). Noncoding regions are the main source of targetable tumor-specific antigens. *Science Translational Medicine*, *10*(470), eaau5516. doi: 10.1126/scitranslmed.aau5516.

Leko, V. and Rosenberg, S. A. (2020). Identifying and targeting human tumor antigens for t cell-based immunotherapy of solid tumors. *Cancer Cell*, *38*(4), 454–472. doi: 10.1016/j.ccell.2020.07.013.

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with bwa-mem.
URL `https://arxiv.org/abs/1303.3997`

Li, S., Schmitz, K. R., Jeffrey, P. D., Wiltzius, J. J. W., Kussie, P., and Ferguson, K. M. (2005). Structural basis for inhibition of the epidermal growth factor receptor by cetuximab. *Cancer Cell*, *7*(4), 301–311. doi: 10.1016/j.ccr.2005.03.003.

Litchfield, K., Stanislaw, S., Spain, L., Gallegos, L. L., Rowan, A., Schnidrig, D., ... Turajlic, S. (2020). Representative sequencing: Unbiased sampling of solid tumor tissue. *Cell Reports*, *31*(5), 107550. doi: 10.1016/j.celrep.2020.107550.

Liu, L. Y., Bhandari, V., Salcedo, A., Espiritu, S. M. G., Morris, Q. D., Kislinger, T. and Boutros, P. C. (2020). Quantifying the influence of mutation detection on tumour subclonal reconstruction. *Nature Communications*, *11*(11), 6247. doi: 10.1038/s41467-020-20055-w.

Lowy, D. R. and Schiller, J. T. (2006). Prophylactic human papillomavirus vaccines. *The Journal of Clinical Investigation*, *116*(5), 1167–1173. doi: 10.1172/JCI28607.

Macklin, A., Khan, S., and Kislinger, T. (2020). Recent advances in mass spectrometry based clinical proteomics: applications to cancer research. *Clinical Proteomics*, *17*(1), 17. doi: 10.1186/s12014-020-09283-w.

Majzner, R. G. and Mackall, C. L. (2018). Tumor antigen escape from car t-cell therapy. *Cancer Discovery*, *8*(10), 1219–1226. doi: 10.1158/2159-8290.CD-18-0442.

Martins, F., Sofiya, L., Sykiotis, G. P., Lamine, F., Maillard, M., Fraga, M., ... Obeid, M. (2019). Adverse effects of immune-checkpoint inhibitors: epidemiology, management and surveillance. *Nature Reviews Clinical Oncology*, *16*(99), 563–580. doi: 10.1038/s41571-019-0218-0.

Martinuzzi, E., Afonso, G., Gagnerault, M.-C., Naselli, G., Mittag, D., Combadière, B., ... Mallone, R. (2011). acDCs enhance human antigen–specific T-cell responses. *Blood*, *118*(8), 2128–2137. doi: 10.1182/blood-2010-12-326231.
URL `https://ashpublications.org/blood/article/118/8/2128/29490/acDCs-enhance-human-antigenspecific-Tcell`

Matsushita, H., Sato, Y., Karasaki, T., Nakagawa, T., Kume, H., Ogawa, S., ... Kakimi, K. (2016). Neoantigen load, antigen presentation machinery, and immune signatures determine prognosis in clear cell renal cell carcinoma. *Cancer Immunology Research*, *4*(5), 463–471. doi: 10.1158/2326-6066.CIR-15-0225.

Maude, S. L., Frey, N., Shaw, P. A., Aplenc, R., Barrett, D. M., Bunin, N. J., ... Grupp, S. A. (2014). Chimeric antigen receptor t cells for sustained remissions in leukemia. *New England Journal of Medicine*, *371*(16), 1507–1517. doi: 10.1056/NEJMoa1407222.

Mazor, T., Pankov, A., Song, J. S., and Costello, J. F. (2016). Intratumoral heterogeneity of the epigenome. *Cancer Cell*, *29*(4), 440–451. doi: 10.1016/j.ccell.2016.03.009.

McGlynn, K. A., Petrick, J. L., and El-Serag, H. B. (2021). Epidemiology of hepatocellular carcinoma. *Hepatology*, *73*(S1), 4–13. doi: 10.1002/hep.31288.

McGranahan, N. and Swanton, C. (2015). Biological and therapeutic impact of intratumor heterogeneity in cancer evolution. *Cancer Cell*, *27*(1), 15–26. doi: 10.1016/j.ccell.2014.12.001.

McGranahan, N. and Swanton, C. (2017). Clonal heterogeneity and tumor evolution: Past, present, and the future. *Cell*, *168*(4), 613–628. doi: 10.1016/j.cell.2017.01.018.

Mellman, I., Coukos, G., and Dranoff, G. (2011). Cancer immunotherapy comes of age. *Nature*, *480*(73787378), 480–489. doi: 10.1038/nature10673.

Michot, J. M., Bigenwald, C., Champiat, S., Collins, M., Carbonnel, F., Postel-Vinay, S., ... Lambotte, O. (2016). Immune-related adverse events with immune checkpoint blockade: a comprehensive review. *European Journal of Cancer*, *54*, 139–148. doi: 10.1016/j.ejca.2015.11.016.

Mikkilineni, L. and Kochenderfer, J. N. (2021). Car t cell therapies for patients with multiple myeloma. *Nature Reviews Clinical Oncology*, *18*(22), 71–84. doi: 10.1038/s41571-020-0427-6.

Moodie, Z., Price, L., Gouttefangeas, C., Mander, A., Janetzki, S., Löwer, M., ... Britten, C. M. (2010). Response definition criteria for elispot assays revisited. *Cancer immunology, immunotherapy: CII*, *59*(10), 1489–1501. doi: 10.1007/s00262-010-0875-4.

Motwani, J. and Eccles, M. R. (2021). Genetic and genomic pathways of melanoma development, invasion and metastasis. *Genes*, *12*(10), 1543. doi: 10.3390/genes12101543.

Moutaftsi, M., Peters, B., Pasquetto, V., Tscharke, D. C., Sidney, J., Bui, H.-H., ... Sette, A. (2006). A consensus epitope prediction approach identifies the breadth of murine TCD8+-cell responses to vaccinia virus. *Nature Biotechnology*, *24*(7), 817–819. doi: 10.1038/nbt1215.
URL `http://www.nature.com/articles/nbt1215`

Mylonas, R., Beer, I., Iseli, C., Chong, C., Pak, H.-S., Gfeller, D., ... Bassani-Sternberg, M. (2018). Estimating the contribution of proteasomal spliced peptides to the hla-i

ligandome*. *Molecular & Cellular Proteomics*, *17*(12), 2347–2357. doi: 10.1074/mcp.RA118.000877.

Navin, N., Kendall, J., Troge, J., Andrews, P., Rodgers, L., McIndoo, J., ... Wigler, M. (2011). Tumour evolution inferred by single-cell sequencing. *Nature*, *472*(73417341), 90–94. doi: 10.1038/nature09807.

Nielsen, M. and Andreatta, M. (2016a). NetMHCpan-3.0; improved prediction of binding to MHC class I molecules integrating information from multiple receptor and peptide length datasets. *Genome Medicine*, *8*(1), 33. doi: 10.1186/s13073-016-0288-x.
URL `http://genomemedicine.biomedcentral.com/articles/10.1186/s13073-016-0288-x`

Nielsen, M. and Andreatta, M. (2016b). Netmhcpan-3.0; improved prediction of binding to mhc class i molecules integrating information from multiple receptor and peptide length datasets. *Genome Medicine*, *8*(1), 33. doi: 10.1186/s13073-016-0288-x.

Nielsen, M., Andreatta, M., Peters, B., and Buus, S. (2020). Immunoinformatics: Predicting peptide–mhc binding. *Annual Review of Biomedical Data Science*, *3*(1), 191–215. doi: 10.1146/annurev-biodatasci-021920-100259.

Nielsen, M., Lundegaard, C., Lund, O., and Keşmir, C. (2005). The role of the proteasome in generating cytotoxic t-cell epitopes: insights obtained from improved predictions of proteasomal cleavage. *Immunogenetics*, *57*(1), 33–41. doi: 10.1007/s00251-005-0781-7.

Nielsen, M., Lundegaard, C., Worning, P., Lauemøller, S. L., Lamberth, K., Buus, S., ... Lund, O. (2003). Reliable prediction of t-cell epitopes using neural networks with novel sequence representations. *Protein Science*, *12*(5), 1007–1017. doi: 10.1110/ps.0239403.

Nieuwenhuis, T. O., Yang, S. Y., Verma, R. X., Pillalamarri, V., Arking, D. E., Rosenberg, A. Z., ... Halushka, M. K. (2020). Consistent rna sequencing contamination in gtex and other data sets. *Nature Communications*, *11*(11), 1933. doi: 10.1038/s41467-020-15821-9.

Noble, P. B. and Cutts, J. H. (1967). Separation of blood leukocytes by ficoll gradient. *The Canadian Veterinary Journal*, *8*(5), 110–111.

O'Donnell, T. J., Rubinsteyn, A., Bonsack, M., Riemer, A. B., Laserson, U., and Hammer-

bacher, J. (2018). MHCflurry: Open-Source Class I MHC Binding Affinity Prediction. *Cell Systems*, *7*(1), 129–132.e4. doi: 10.1016/j.cels.2018.05.014.
URL https://linkinghub.elsevier.com/retrieve/pii/S2405471218302321

Ott, P. A., Hu, Z., Keskin, D. B., Shukla, S. A., Sun, J., Bozym, D. J., ... Wu, C. J. (2017). An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature*, *547*(76627662), 217–221. doi: 10.1038/nature22991.

Ouspenskaia, T., Law, T., Clauser, K. R., Klaeger, S., Sarkizova, S., Aguet, F., ... Regev, A. (2022). Unannotated proteins expand the mhc-i-restricted immunopeptidome in cancer. *Nature Biotechnology*, *40*(22), 209–217. doi: 10.1038/s41587-021-01021-3.

O'Donnell, J. S., Teng, M. W. L., and Smyth, M. J. (2019). Cancer immunoediting and resistance to t cell-based immunotherapy. *Nature Reviews Clinical Oncology*, *16*(33), 151–167. doi: 10.1038/s41571-018-0142-8.

O'Donnell, T. J., Rubinsteyn, A., Bonsack, M., Riemer, A. B., Laserson, U., and Hammerbacher, J. (2018). Mhcflurry: Open-source class i mhc binding affinity prediction. *Cell Systems*, *7*(1), 129–132.e4. doi: 10.1016/j.cels.2018.05.014.

O'Malley, D., Lee, S., Psyrri, A., Sukari, A., Thomas, S., Wenham, R., ... Jimeno, A. (2021). 492 phase 2 efficacy and safety of autologous tumor-infiltrating lymphocyte (til) cell therapy in combination with pembrolizumab in immune checkpoint inhibitor-naïve patients with advanced cancers. *Journal for ImmunoTherapy of Cancer*, *9*(Suppl 2). doi: 10.1136/jitc-2021-SITC2021.492.
URL https://jitc.bmj.com/content/9/Suppl_2/A523

Pardoll, D. M. (2012). The blockade of immune checkpoints in cancer immunotherapy. *Nature Reviews Cancer*, *12*(44), 252–264. doi: 10.1038/nrc3239.

Parkhurst, M. R., Robbins, P. F., Tran, E., Prickett, T. D., Gartner, J. J., Jia, L., ... Rosenberg, S. A. (2019). Unique neoantigens arise from somatic mutations in patients with gastrointestinal cancers. *Cancer Discovery*, *9*(8), 1022–1035. doi: 10.1158/2159-8290.CD-18-1494.

Paul, S., Weiskopf, D., Angelo, M. A., Sidney, J., Peters, B., and Sette, A. (2013). HLA Class

I Alleles Are Associated with Peptide-Binding Repertoires of Different Size, Affinity, and Immunogenicity. *The Journal of Immunology*, *191*(12), 5831–5839. doi: 10.4049/jimmunol.1302101.
URL http://www.jimmunol.org/lookup/doi/10.4049/jimmunol.1302101

Pearson, H., Daouda, T., Granados, D. P., Durette, C., Bonneil, E., Courcelles, M., ... Perreault, C. (2016). Mhc class i–associated peptides derive from selective regions of the human genome. *The Journal of Clinical Investigation*, *126*(12), 4690–4701. doi: 10.1172/JCI88590.

Peng, X., Xu, X., Wang, Y., Hawke, D. H., Yu, S., Han, L., ... Mills, G. B. (2018). A-to-i rna editing contributes to proteomic diversity in cancer. *Cancer Cell*, *33*(5), 817–828.e7. doi: 10.1016/j.ccell.2018.03.026.

Peters, B., Bui, H.-H., Frankild, S., Nielsen, M., Lundegaard, C., Kostem, E., ... Sette, A. (2006). A community resource benchmarking predictions of peptide binding to mhc-i molecules. *PLOS Computational Biology*, *2*(6), e65. doi: 10.1371/journal.pcbi.0020065.

Pierpont, T. M., Limper, C. B., and Richards, K. L. (2018). Past, present, and future of rituximab—the world's first oncology monoclonal antibody therapy. *Frontiers in Oncology*, *8*.
URL https://www.frontiersin.org/articles/10.3389/fonc.2018.00163

Pimentel Muniz, T., Sorotsky, H., Kanjanapan, Y., Rose, A. A. N., Araujo, D. V., Fortuna, A., ... Hogg, D. (2020). Genomic landscape of malignant peripheral nerve sheath tumor (mpnst)-like melanoma. *Journal of Clinical Oncology*, *38*(15_suppl), e22072–e22072. doi: 10.1200/JCO.2020.38.15_suppl.e22072.

Powles, T., Rosenberg, J. E., Sonpavde, G. P., Loriot, Y., Durán, I., Lee, J.-L., ... Petrylak, D. P. (2021). Enfortumab vedotin in previously treated advanced urothelial carcinoma. *New England Journal of Medicine*, *384*(12), 1125–1135. doi: 10.1056/NEJMoa2035807.

Purcell, A. W., Ramarathinam, S. H., and Ternette, N. (2019). Mass spectrometry–based identification of mhc-bound peptides for immunopeptidomics. *Nature Protocols*, *14*(66), 1687–1707. doi: 10.1038/s41596-019-0133-y.

Rafiq, S., Hackett, C. S., and Brentjens, R. J. (2020). Engineering strategies to overcome the current roadblocks in car t cell therapy. *Nature Reviews Clinical Oncology*, *17*(33), 147–167. doi: 10.1038/s41571-019-0297-y.

Raje, N., Berdeja, J., Lin, Y., Siegel, D., Jagannath, S., Madduri, D., ... Kochenderfer, J. N. (2019). Anti-bcma car t-cell therapy bb2121 in relapsed or refractory multiple myeloma. *New England Journal of Medicine*, *380*(18), 1726–1737. doi: 10.1056/NEJMoa1817226.

Rammensee, H. G., Bachmann, J., Emmerich, N. P. N., Bachor, O. A., and Stevanović, S. (1999). Syfpeithi: database for mhc ligands and peptide motifs. *Immunogenetics*, *50*(3), 213–219. doi: 10.1007/s002510050595.

Ramos, C. A., Grover, N. S., Beaven, A. W., Lulla, P. D., Wu, M.-F., Ivanova, A., ... Savoldo, B. (2020). Anti-cd30 car-t cell therapy in relapsed and refractory hodgkin lymphoma. *Journal of Clinical Oncology*, *38*(32), 3794–3804. doi: 10.1200/JCO.20.01342.

Ranieri, E., Popescu, I., and Gigante, M. (2014). *CTL ELISPOT Assay*, (pp. 75–86). New York, NY: Springer New York. doi: 10.1007/978-1-4939-1158-5_6.
URL https://doi.org/10.1007/978-1-4939-1158-5_6

Riley, R. S., June, C. H., Langer, R., and Mitchell, M. J. (2019). Delivery technologies for cancer immunotherapy. *Nature Reviews Drug Discovery*, *18*(33), 175–196. doi: 10.1038/s41573-018-0006-z.

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

Rotte, A., Jin, J. Y., and Lemaire, V. (2018). Mechanistic overview of immune checkpoints to support the rational design of their combinations in cancer immunotherapy. *Annals of Oncology*, *29*(1), 71–83. doi: 10.1093/annonc/mdx686.

Sade-Feldman, M., Jiao, Y. J., Chen, J. H., Rooney, M. S., Barzily-Rokni, M., Eliane, J.-P., ... Hacohen, N. (2017). Resistance to checkpoint blockade therapy through inactivation of antigen presentation. *Nature Communications*, *8*(11), 1136. doi: 10.1038/s41467-017-01062-w.

Sahin, U. and Türeci, O. (2018). Personalized vaccines for cancer immunotherapy. *Science*, *359*(6382), 1355–1360. doi: 10.1126/science.aar7112.

Salles, G., Barrett, M., Foà, R., Maurer, J., O'Brien, S., Valente, N., ... Maloney, D. G. (2017). Rituximab in b-cell hematologic malignancies: A review of 20 years of clinical experience. *Advances in Therapy*, *34*(10), 2232–2273. doi: 10.1007/s12325-017-0612-x.

Sanli, Y., Leake, J., Odu, A., Xi, Y., and Subramaniam, R. M. (2019). Tumor heterogeneity on fdg pet/ct and immunotherapy: An imaging biomarker for predicting treatment response in patients with metastatic melanoma. *American Journal of Roentgenology*, *212*(6), 1318–1326. doi: 10.2214/AJR.18.19796.

Saxena, M., van der Burg, S. H., Melief, C. J. M., and Bhardwaj, N. (2021). Therapeutic cancer vaccines. *Nature Reviews Cancer*, *21*(66), 360–378. doi: 10.1038/s41568-021-00346-0.

Schnell, A., Bod, L., Madi, A., and Kuchroo, V. K. (2020). The yin and yang of co-inhibitory receptors: toward anti-tumor immunity without autoimmunity. *Cell Research*, *30*(44), 285–299. doi: 10.1038/s41422-020-0277-x.

Schoenfeld, A., Lee, S., Paz-Ares, L., Doger, B., Gettinger, S., Haefliger, S., ... He, K. (2021). 458 first phase 2 results of autologous tumor-infiltrating lymphocyte (til; ln-145) monotherapy in patients with advanced, immune checkpoint inhibitor-treated, non-small cell lung cancer (nsclc). *Journal for ImmunoTherapy of Cancer*, *9*(Suppl 2). doi: 10.1136/jitc-2021-SITC2021.458.
URL `https://jitc.bmj.com/content/9/Suppl_2/A486`

Schumacher, T. N., Scheper, W., and Kvistborg, P. (2019). Cancer neoantigens. *Annual Review of Immunology*, *37*(1), 173–200. doi: 10.1146/annurev-immunol-042617-053402.

Schumacher, T. N. and Schreiber, R. D. (2015). Neoantigens in cancer immunotherapy. *Science*, *348*(6230), 69–74. doi: 10.1126/science.aaa4971.

Scott, A. M., Wolchok, J. D., and Old, L. J. (2012). Antibody therapy of cancer. *Nature Reviews Cancer*, *12*(44), 278–287. doi: 10.1038/nrc3236.

Shen, L., Zhang, J., Lee, H., Batista, M. T., and Johnston, S. A. (2019). Rna transcription

and splicing errors as a source of cancer frameshift neoantigens for vaccines. *Scientific Reports*, *9*(1), 14184. doi: 10.1038/s41598-019-50738-4.

Sherry, S. T., Ward, M., and Sirotkin, K. (1999). dbsnp—database for single nucleotide polymorphisms and other classes of minor genetic variation. *Genome research*, *9*(8), 677–679.

Shi, W., Ng, C. K. Y., Lim, R. S., Jiang, T., Kumar, S., Li, X., ... Hatzis, C. (2018). Reliability of whole-exome sequencing for assessing intratumor genetic heterogeneity. *Cell Reports*, *25*(6), 1446–1457. doi: 10.1016/j.celrep.2018.10.046.

Simpson, A. J. G., Caballero, O. L., Jungbluth, A., Chen, Y.-T., and Old, L. J. (2005). Cancer/testis antigens, gametogenesis and cancer. *Nature Reviews Cancer*, *5*(88), 615–625. doi: 10.1038/nrc1669.

Smith, J. G., Liu, X., Kaufhold, R. M., Clair, J., and Caulfield, M. J. (2001). Development and validation of a gamma interferon elispot assay for quantitation of cellular immune responses to varicella-zoster virus. *Clinical Diagnostic Laboratory Immunology*, *8*(5), 871–879. doi: 10.1128/CDLI.8.5.871-879.2001.

Smyth, M. J., Dunn, G. P., and Schreiber, R. D. (2006). *Cancer Immunosurveillance and Immunoediting: The Roles of Immunity in Suppressing Tumor Development and Shaping Tumor Immunogenicity*, vol. 90 of *Cancer Immunotherapy*, (p. 1–50). Academic Press. doi: 10.1016/S0065-2776(06)90001-7.
URL https://www.sciencedirect.com/science/article/pii/S0065277606900017

Socinski, M. A., Jotte, R. M., Cappuzzo, F., Orlandi, F., Stroyakovskiy, D., Nogami, N., ... Reck, M. (2018). Atezolizumab for first-line treatment of metastatic nonsquamous nsclc. *New England Journal of Medicine*, *378*(24), 2288–2301. doi: 10.1056/NEJMoa1716948.

Sterner, R. C. and Sterner, R. M. (2021). Car-t cell therapy: current limitations and potential strategies. *Blood Cancer Journal*, *11*(44), 1–11. doi: 10.1038/s41408-021-00459-7.

Sullivan, L. A. and Brekken, R. A. (2010). The vegf family in cancer and antibody-based strategies for their inhibition. *mAbs*, *2*(2), 165–175. doi: 10.4161/mabs.2.2.11360.

Szolek, A., Schubert, B., Mohr, C., Sturm, M., Feldhahn, M., and Kohlbacher, O. (2014). Optitype: precision hla typing from next-generation sequencing data. *Bioinformatics*, *30*(23), 3310–3316. doi: 10.1093/bioinformatics/btu548.

Tagliamento, M., Agostinetto, E., Borea, R., Brandão, M., Poggio, F., Addeo, A. and Lambertini, M. (2021). Vista: A promising target for cancer immunotherapy? *ImmunoTargets and Therapy*, *10*, 185–200. doi: 10.2147/ITT.S260429.

Tan, M. H., Li, Q., Shanmugam, R., Piskol, R., Kohler, J., Young, A. N., ... Li, J. B. (2017). Dynamic landscape and regulation of rna editing in mammals. *Nature*, *550*(76757675), 249–254. doi: 10.1038/nature24041.

Tran, E., Ahmadzadeh, M., Lu, Y.-C., Gros, A., Turcotte, S., Robbins, P. F., ... Rosenberg, S. A. (2015). Immunogenicity of somatic mutations in human gastrointestinal cancers. *Science (New York, N.Y.)*, *350*(6266), 1387–1390. doi: 10.1126/science.aad1253.

Tretter, C., de Andrade Krätzig, N., Pecoraro, M., Lange, S., Seifert, P., von Frankenberg, C., ... Krackhardt, A. M. (2023). Proteogenomic analysis reveals rna as a source for tumor-agnostic neoantigen identification. *Nature Communications*, (p. to appear).

Trolle, T., McMurtrey, C. P., Sidney, J., Bardet, W., Osborn, S. C., Kaever, T., ... Peters, B. (2016). The length distribution of class i–restricted t cell epitopes is determined by both peptide supply and mhc allele–specific binding preference. *The Journal of Immunology*, *196*(4), 1480–1487. doi: 10.4049/jimmunol.1501721.

Tumeh, P. C., Harview, C. L., Yearley, J. H., Shintaku, I. P., Taylor, E. J. M., Robert, L., ... Ribas, A. (2014). Pd-1 blockade induces responses by inhibiting adaptive immune resistance. *Nature*, *515*(75287528), 568–571. doi: 10.1038/nature13954.

Tyanova, S., Temu, T., and Cox, J. (2016). The MaxQuant computational platform for mass spectrometry-based shotgun proteomics. *Nature Protocols*, *11*(12), 2301–2319. doi: 10.1038/nprot.2016.136.
URL `http://www.nature.com/articles/nprot.2016.136`

Uhlen, M., Zhang, C., Lee, S., Sjöstedt, E., Fagerberg, L., Bidkhori, G. ... (2017). A pathology atlas of the human cancer transcriptome. *Science*, *357*(6352), eaan2507.

Villani, A.-C., Sarkizova, S., and Hacohen, N. (2018). Systems immunology: Learning the rules of the immune system. *Annual Review of Immunology*, *36*(1), 813–842. doi: 10.1146/annurev-immunol-042617-053035.

Vita, R., Overton, J. A., Greenbaum, J. A., Ponomarenko, J., Clark, J. D., Cantrell, J. R., ... Peters, B. (2015). The immune epitope database (iedb) 3.0. *Nucleic Acids Research*, *43*(D1), D405–D412. doi: 10.1093/nar/gku938.

Vitiello, A. and Zanetti, M. (2017). Neoantigen prediction and the need for validation. *Nature Biotechnology*, *35*(99), 815–817. doi: 10.1038/nbt.3932.

von Minckwitz, G., Huang, C.-S., Mano, M. S., Loibl, S., Mamounas, E. P., Untch, M., ... Geyer, C. E. (2019). Trastuzumab emtansine for residual invasive her2-positive breast cancer. *New England Journal of Medicine*, *380*(7), 617–628. doi: 10.1056/NEJMoa1814017.

Waldman, A. D., Fritz, J. M., and Lenardo, M. J. (2020). A guide to cancer immunotherapy: from t cell basic science to clinical practice. *Nature Reviews Immunology*, *20*(1111), 651–668. doi: 10.1038/s41577-020-0306-5.

Wang, M., Munoz, J., Goy, A., Locke, F. L., Jacobson, C. A., Hill, B. T., ... Reagan, P. M. (2020). Kte-x19 car t-cell therapy in relapsed or refractory mantle-cell lymphoma. *New England Journal of Medicine*, *382*(14), 1331–1342. doi: 10.1056/NEJMoa1914347.

Weinmann, S. C. and Pisetsky, D. S. (2019). Mechanisms of immune-related adverse events during the treatment of cancer with immune checkpoint inhibitors. *Rheumatology*, *58*(Supplement_7), vii59–vii67. doi: 10.1093/rheumatology/kez308.

Wells, D. K., van Buuren, M. M., Dang, K. K., Hubbard-Lucey, V. M., Sheehan, K. C. F., Campbell, K. M., ... Defranoux, N. A. (2020). Key parameters of tumor epitope immunogenicity revealed through a consortium approach improve neoantigen prediction. *Cell*, *183*(3), 818–834.e13. doi: 10.1016/j.cell.2020.09.015.

Wieczorek, M., Abualrous, E. T., Sticht, J., Álvaro Benito, M., Stolzenberg, S., Noé, F. and Freund, C. (2017). Major histocompatibility complex (mhc) class i and mhc class ii proteins: Conformational plasticity in antigen presentation. *Frontiers in Immunology*, *8*. URL https://www.frontiersin.org/articles/10.3389/fimmu.2017.00292

Xie, C., Yeo, Z. X., Wong, M., Piper, J., Long, T., Kirkness, E. F., ... 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*, *114*(30), 8059–8064. doi: 10.1073/pnas.1707945114.

Xie, N., Shen, G., Gao, W., Huang, Z., Huang, C., and Fu, L. (2023). Neoantigens: promising targets for cancer therapy. *Signal Transduction and Targeted Therapy*, *8*(11), 1–38. doi: 10.1038/s41392-022-01270-x.

Xie, Y., Liu, C., Zhao, Y., Gong, C., Li, Y., Hu, S., ... Wang, B. (2022). Heterogeneity derived from 18f-fdg pet/ct predicts immunotherapy outcome for metastatic triple-negative breast cancer patients. *Cancer Medicine*, *11*(9), 1948–1955. doi: 10.1002/cam4.4522.

Xu, X., Wang, Y., and Liang, H. (2018). The role of a-to-i rna editing in cancer development. *Current Opinion in Genetics & Development*, *48*, 51–56. doi: 10.1016/j.gde.2017.10.009.

Yadav, M., Jhunjhunwala, S., Phung, Q. T., Lupardus, P., Tanguay, J., Bumbaca, S., ... Delamarre, L. (2014). Predicting immunogenic tumour mutations by combining mass spectrometry and exome sequencing. *Nature*, *515*(75287528), 572–576. doi: 10.1038/nature14001.

Yarchoan, M., Johnson, B. A., Lutz, E. R., Laheru, D. A., and Jaffee, E. M. (2017). Targeting neoantigens to augment antitumour immunity. *Nature Reviews Cancer*, *17*(44), 209–222. doi: 10.1038/nrc.2016.154.

Yewdell, J. W. and Bennink, J. R. (1999). Immunodominance in major histocompatibility complex class i–restricted t lymphocyte responses. *Annual Review of Immunology*, *17*(1), 51–88. doi: 10.1146/annurev.immunol.17.1.51.

Zahavi, D. and Weiner, L. (2020). Monoclonal antibodies in cancer therapy. *Antibodies*, *9*(33), 34. doi: 10.3390/antib9030034.

Zaidi, N., Soban, M., Chen, F., Kinkead, H., Mathew, J., Yarchoan, M., ... Jaffee, E. M. (2020). Role of in silico structural modeling in predicting immunogenic neoepitopes for cancer vaccine development. *JCI Insight*, *5*(17), e136991. doi: 10.1172/jci.insight.136991.

Zareie, P., Szeto, C., Farenc, C., Gunasinghe, S. D., Kolawole, E. M., Nguyen, A., ... La Gruta, N. L. (2021). Canonical t cell receptor docking on peptide–mhc is essential for t cell signaling. *Science*, *372*(6546), eabe9124. doi: 10.1126/science.abe9124.

Zehir, A., Benayed, R., Shah, R. H., Syed, A., Middha, S., Kim, H. R., ... Berger, M. F. (2017). Mutational landscape of metastatic cancer revealed from prospective clinical sequencing of 10,000 patients. *Nature Medicine*, *23*(66), 703–713. doi: 10.1038/nm.4333.

Zhang, X., Qi, Y., Zhang, Q., and Liu, W. (2019). Application of mass spectrometry-based mhc immunopeptidome profiling in neoantigen identification for tumor immunotherapy. *Biomedicine & Pharmacotherapy*, *120*, 109542. doi: 10.1016/j.biopha.2019.109542.

Zhao, W. and Sher, X. (2018). Systematically benchmarking peptide-MHC binding predictors: From synthetic to naturally processed epitopes. *PLOS Computational Biology*, *14*(11), e1006457. doi: 10.1371/journal.pcbi.1006457.
URL http://dx.plos.org/10.1371/journal.pcbi.1006457

Zhou, C., Wei, Z., Zhang, L., Yang, Z., and Liu, Q. (2020). Systematically characterizing a-to-i rna editing neoantigens in cancer. *Frontiers in Oncology*, *10*.
URL https://www.frontiersin.org/articles/10.3389/fonc.2020.593989

Appendix A: Source code

1 Scripts in Python

1.1 Run netMHC

```
1
2
3  import csv # module for reading and writing .csv-file
4  import os
5
6  # REMINDER: out commented version: debug mode for linux server/macOS.
7
8  folder_input = "/root/netMHC/input_files/mut_peptides_3/"
9  #folder_input = "/Users/.../input_files/mut_peptides/
       preselected_for_netMHC/"
10 folder_output = "/root/netMHC/output_files/mut_peptides_3/"
11
12 # read .csv file and create array "alleles"
13
14 with open(folder_input+'alleles.csv') as csvfile:
15   reader = csv.reader(csvfile, delimiter=' ', quotechar='|')
16   alleles = list(csv.reader(csvfile))
17
18 # iterate to delete all NAs
19 for i in range(len(alleles[0][1:])):
20   for k in alleles:
21      try:
22         k.remove('NA')
23      except ValueError:
24         pass
25
26 patients=[]
27 for i in range(len(alleles)):
28   patients.append(alleles[i][0])
29
30 for x in range(len(patients)):
```

```
31  os.system("netMHC -a " + ','.join(alleles[x][1:])+" -l
        8,9,10,11,12,13,14,15 -xls -xlsfile " + folder_output + "result_"
         + patients[x] + "_netMHC.tsv " + folder_input + patients[x] + ".
        fasta > " + folder_output + "result_" + patients[x] + "_netMHC.
        csv")
32  # print("netMHC -a " + ','.join(alleles[x][1:])+" -l
        8,9,10,11,12,13,14,15 -xls -xlsfile " + folder_output + "result_"
         + patients[x] + "_netMHC.tsv " + folder_input + patients[x] + ".
        fasta > " + folder_output + "result_" + patients[x] + "_netMHC.
        csv")
```

Listing 1: Implementation of netMHC

1.2 Run MHCflurry

```
1  import csv # module for writing .csv-file
2  import os
3
4  folder_all = "/Users/.../Python/input_files/mut_peptides/all"
5
6  from os import walk
7
8  g = []
9  pep_nrs_C = []
10
11 for (dirpath, dirnames, filenames) in walk(folder_all):
12    g.extend(filenames)
13    break
14
15 for j in range(len(g)):
16   pep_nrs_C.append(g[j][3:5])
17   print(j+1, "in All: " + g[j][0:5])
18
19 n_patients_C = len(pep_nrs_C)
20
21 command_1 = ("mhcflurry-predict /Users/.../Python/input_files/
       mut_peptides/all/IN_")
22 command_2 = (".csv --out /Users/.../Python/result_files/mut_peptides/")
23 command_3 = ("models_class1_presentation/result_")
24 command_4 = ("_st.csv")
25
26 for p in range(n_patients_C):
27   # standard model (class1)
```

```
28 command_st = command_1 + str(pep_nrs_C[p]) + command_2 + command_3 +
       str(pep_nrs_C[p]) + command_4
29 os.system(command_st)
30 print(command_st)
```

Listing 2: Implementation of MHCflurry

2 Scripts in R

2.1 Characterization of the mutanome

```
#### MUTATION ANALYSIS PLOT FOR THESIS####

library(tidyverse)
library(openxlsx)
library(stringr)
library(reshape2)
library(ggplot2)
library(extrafont)
library(ggrepel)
library(grid)
library(FinCal)
library(gridExtra)
library(RColorBrewer)
library(grDevices)
library(ggpubr)
library(cowplot)
library(remotes)
library(webr)
library(moonBook)
library(plotly)
library(ggrepel)
library(ggbreak)
library(hrbrthemes)
library(viridis)
library(ggExtra)
library(eulerr)
library(ggh4x)


#### (1) #### import all .tsv files from folder and add to one dataframe ____________________________________________________________________________________________
path.tsv.files="mutation_calling/raw_data/2022_07_20/"

# old: path.tsv.files="mutation_calling/2019_06_05/results_out/"
tsv.files=list.files(path=path.tsv.files, pattern = "*.tsv", full.names = T)
all.patients_DF = plyr::ldply(tsv.files, read.delim)# fread

#### (2) #### correct TumorVF & NormalVF ____________________________________________
```

```
    ------------------------------------------------------------------------
    ----------
all.patients_DF <- mutate(all.patients_DF, TumorVF=TumorAD/(TumorAD+
    TumorRD))
all.patients_DF <- mutate(all.patients_DF, NormalVF=NormalAD/(NormalAD+
    NormalRD))

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

all.patients_DF <- all.patients_DF %>%
 rename(Master_ID=patientID)
all.patients_DF$Master_ID_group <- as.factor(str_sub(all.patients_DF$
    Master_ID,1,6))
all.patients_DF <- merge(reference.master, all.patients_DF, 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"
```

```
                                    ,
62                                  "stop_gained"="Stop gained",
63                                  "splice_donor_variant&intron_variant"="
                                        Splice donor variant & intron
                                        variant",
64                                  "frameshift_variant"="Frameshift
                                        variant",
65                                  "disruptive_inframe_deletion"="
                                        Disruptive inframe deletion",
66                                  "splice_acceptor_variant&intron_variant
                                        "="Splice acceptor variant & intron
                                         variant",
67                                  "Splice donor variant&intron_variant"="
                                        Splice donor variant & intron
                                        variant",
68                                  "Splice acceptor variant&intron_variant
                                        "="Splice acceptor variant & intron
                                         variant"))) %>%
69  mutate(geneBiotype=str_replace_all(geneBiotype, c("3prime_overlapping
        _ncRNA"="3'-overlapping ncRNA",
70                                      "antisense" = "Antisense",
71                                      "processed_pseudogene"="
                                            Processed Pseudogene",
72                                      "protein_coding"="Protein Coding
                                            ",
73                                      "transcribed_Processed
                                            Pseudogene"="Processed
                                            Pseudogene (transcribed)",
74                                      "unProcessed Pseudogene"="
                                            Unprocessed Pseudogene",
75                                      "sense_intronic"="Sense Intronic
                                            ",
76                                      "transcribed_Unprocessed
                                            Pseudogene;processed_
                                            transcript"="Unprocessed
                                            Pseudogene (transcribed) +
                                            pt",
77                                      "transcribed_Unprocessed
                                            Pseudogene"="Unprocessed
                                            Pseudogene (transcribed)",
78                                      "unitary_pseudogene"="Unitary
                                            Pseudogene",
79                                      "processed_transcript"="
                                            Processed Transcript",
80                                      "IG_V_pseudogene"="Variable
                                            chain IG Pseudogene",
```

```
                                        "sense_overlapping"="Sense
                                            overlapping")))

### (4) ### General modifications of dataset ____________________________
    ____________________________________________________________________
    ____________________________________________________________________
    _____________

all.patients_DF["Tumor_ID"] <- paste0(all.patients_DF$Patient_ID, str_
    sub(all.patients_DF$Master_ID, 7,9))
all.patients_DF["Mutation_ID"] <- paste(all.patients_DF$CHROM, all.
    patients_DF$POS,all.patients_DF$REF, all.patients_DF$ALT, sep = "_"
    )
all.patients_DF["Metastasis"] <- str_sub(all.patients_DF$Master_ID,
    8,9)
all.patients_DF["Patient_NR"] <- paste0(str_sub(all.patients_DF$Patient
    _ID, 4,5))
all.patients_DF <- all.patients_DF %>%
 mutate(Tumor_entity_short_simple=ifelse(all.patients_DF$Tumor_entity_
     short=="Melanoma", "Melanoma", ifelse(all.patients_DF$Tumor_
     entity_short=="Sarcoma", "Sarcoma", ifelse(all.patients_DF$Tumor_
     entity_short=="Carcinoma", "Carcinoma", "Other")))) %>%
 mutate(filter5=ifelse(TumorVF>0.05 & (TumorAD+TumorRD)>4 & (TumorAD
     >2), "Inliers", "Outliers")) # Filter for TumorVF > 5% and
     minimal coverage of reads of 5 and minimal tumor reads of 3

## create new column Biotype_group
all.patients_DF <- all.patients_DF %>%
 mutate(Biotype_group=geneBiotype) %>%
 mutate(Biotype_group=ifelse(grepl("pseudogene", geneBiotype, ignore.
     case = T), "Pseudogene", Biotype_group)) %>%
 mutate(Biotype_group=ifelse(geneBiotype %in% c("bidirectional_
     promoter_lncRNA", "macro_lncRNA"), "lncRNA", Biotype_group)) %>%
 mutate(Biotype_group=ifelse(geneBiotype %in% c("Sense overlapping", "
     IG_V_gene", "IG_C_gene", "TR_V_gene", "non_coding"), "Others",
     Biotype_group)) %>%
 mutate(Biotype_group=ifelse(Biotype_group %in% c("3'-overlapping
     ncRNA", "misc_RNA", "lncRNA", "snoRNA", "snRNA", "miRNA", "scaRNA
     ", "rRNA", "vaultRNA", "lincRNA", "Antisense"), "Regulatory RNAs"
     , Biotype_group))

## Debug check occurrence of each biotype
test <- group_by(all.patients_DF, Biotype_group) %>%
 summarise(N=n()) %>%
 arrange(desc(N))
test
```

```
106
107 ## create new column Effect_group
108 all.patients_DF <- all.patients_DF %>%
109   mutate(Effect_group=EFFECT) %>%
110   mutate(Effect_group=ifelse(grepl("exon", EFFECT, ignore.case = T), "
        Non-coding", Effect_group)) %>%
111   mutate(Effect_group=ifelse(grepl("missense", EFFECT, ignore.case = T)
        , "Coding missense", Effect_group)) %>%
112   mutate(Effect_group=ifelse(grepl("splice", EFFECT, ignore.case = T),
        "Splice site & intron", Effect_group)) %>%
113   mutate(Effect_group=ifelse(grepl("stop", EFFECT, ignore.case = T), "
        Stop gained", Effect_group)) %>%
114   mutate(Effect_group=ifelse(grepl("Frameshift", EFFECT, ignore.case =
        T), "Frameshift", Effect_group)) %>%
115   mutate(Effect_group=ifelse(Effect_group %in% c("Non-coding","Coding
        missense", "Splice site & intron", "Stop gained", "Frameshift"),
        Effect_group, "Others"))
116
117 ## Debug check occurrence of each Effect_group
118 test <- group_by(all.patients_DF, Effect_group)%>%
119   summarise(N=n()) %>%
120   arrange(desc(N))
121 test
122
123 ### (5) ### For Variant plot ____________________________________________
      ____________________________________________________________________
      ____________________________________________________________________
124
125 variants <- all.patients_DF
126 variants <- variants %>%
127   mutate(Patient_NR_plot=ifelse(variants$Metastasis=="T2"|variants$
        Metastasis=="T4", NA, Patient_NR)) %>%
128   mutate(Metastasis_plot=ifelse(variants$Patient_NR=="19"|variants$
        Patient_NR=="11"|variants$Patient_NR=="17"|variants$Patient_NR=="
        23"|variants$Patient_NR=="24"|variants$Patient_NR=="25"|variants$
        Patient_NR=="27", Metastasis, "")) %>%
129   mutate(Pat_Met=paste0(Patient_NR, "_", Metastasis))
130
131 variants_unique <- variants %>%
132   distinct(Tumor_ID, Mutation_ID, SOURCE, GENE, .keep_all=T) %>% #
        Mainly differences in ORF are cut off, but we want to keep the
        GENE differences
133   group_by(Tumor_ID, Mutation_ID, SOURCE) %>%
134   summarise(across(!c(GENE), first), GENE=paste0(GENE, collapse = ";"))
        # vermutlich elegante Loesung, aber nicht sehr Rechen-effizient
135
```

```
136 # Save Plot
137 save.path = "plots/Thesis/Variants/"
138 speichern <- function(name){
139   save.filename = paste(save.path,name, ".pdf", sep = "")
140   ggsave(save.filename, width = 20, height = 10, dpi = "retina")
141 }
142
143 speichern_grid <- function(name){
144   save.filename = paste(save.path,name, ".pdf", sep = "")
145   ggsave(save.filename, g, width = 20, height = 10, dpi = "retina")
146 }
147
148 ### (6) ### Themes ____________________________________________________
      ____________________________________________________________________
      ____________________________________
149
150 theme_donut <- function(){
151   theme(aspect.ratio = 1,
152         strip.text = element_text(size = 30),
153         legend.key.size = unit(2.5, 'cm'),
154         legend.title = element_text(size=30),
155         legend.text = element_text(size=30))
156 }
157
158 theme_basic <- function(){
159   theme_minimal()+
160   theme(axis.title = element_text(size=30),
161         axis.text = element_text(size=30),
162         axis.text.y.right = element_blank(),
163         axis.ticks.x=element_blank(),
164         panel.grid.major.x=element_blank(),
165         panel.grid.minor=element_blank(),
166         legend.key.size = unit(2.5, 'cm'),
167         legend.title = element_text(size=30),
168         legend.text = element_text(size=30),
169         strip.text.y = element_text(size = 30))
170 }
171
172 theme_PS <- function(){
173   theme(
174     plot.title=element_text(size=20, hjust = 0.5),
175     #plot.background = element_rect(fill = "transparent",colour = NA),
176     #panel.grid.major = element_line(color = "grey", linetype = "dotted
          ", size=0.4),
177     panel.grid.major = element_blank(),
178     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 = 16,face="bold"),
    legend.text = element_text(size = 16),
    legend.title = element_text(size= 16,face="bold")
  )
}

### PLOT 1b ### ____________________________________________________________
    ____________________________________________________________________
    ____________________________________

overview1 <- variants %>%
  distinct(SOURCE, Patient_ID, Patient_NR, Master_ID, CHROM, POS, REF,
      ALT, .keep_all = T) %>%
  group_by(SOURCE, mutationType) %>%
  summarise(count_variants=n()) %>%
  rename("Variants"=SOURCE)
levels(overview2$Variants) <- c("DNA", "RNA")

PieDonut(overview1, aes(Variants, mutationType, count=count_variants),
       pieAlpha = 1.0, donutAlpha = 1.0, color = "black",
       labelpositionThreshold=0.05, labelposition = 1,
       ratioByGroup=T, start=0,
       r0=0.4,r1=1.2,r2=2.0)

overview2 <- variants %>%
  distinct(SOURCE, Patient_ID, Patient_NR, Master_ID, CHROM, POS, REF,
      ALT, .keep_all = T) %>%
  group_by(SOURCE, filter5) %>%
  summarise(count_variants=n()) %>%
  rename("Variants"=SOURCE) %>%
  ungroup() %>%
  mutate(filter5=as.factor(filter5))
levels(overview2$Variants) <- c("DNA", "RNA")

ggplot(overview2, aes(x = filter5, y = count_variants, fill = Variants)
    ) +
  geom_col() +
  scale_fill_viridis_d()
  #coord_polar("y")

```

```
PieDonut(overview2, aes(Variants, filter5, count=count_variants),
       pieAlpha = 1.0, donutAlpha = 1.0, color = "black",
       labelpositionThreshold=0.05, labelposition = 0,
       ratioByGroup=T, start=0,
       r0=0.4,r1=1.2,r2=2.0)

grid.newpage()
g <- grid.arrange(p1, p21)

hsize <- 2

overview2 <- variants %>%
  distinct(SOURCE, Patient_ID, Patient_NR, Master_ID, CHROM, POS, REF,
      ALT, .keep_all = T) %>%
  group_by(SOURCE, filter5) %>%
  summarise(count_variants=n()) %>%
  rename("Variants"=SOURCE) %>%
  mutate(x = hsize)
levels(overview2$Variants) <- c("DNA", "RNA")

ggplot(overview1, aes(x = hsize, y = count_variants, fill =
    mutationType)) +
  geom_col(color = "black") +
  geom_text(aes(label = count_variants),
          position = position_stack(vjust = 0.5)) +
  coord_polar(theta = "y") +
  scale_fill_brewer(palette = "GnBu") +
  xlim(c(0.2, hsize + 0.5)) +
  theme(panel.background = element_rect(fill = "white"),
       panel.grid = element_blank(),
       axis.title = element_blank(),
       axis.ticks = element_blank(),
       axis.text = element_blank())

#creating simulated data
data <- data.frame(c('Solar Panels', 'Fossil Fuels', 'Wind Turbines'),
    c(15,80,5), c(40, 40, 20))
colnames(data) <- c('Energy Source', 'United States', 'European Union')

plot3 <- plot_ly(data) %>%
  add_pie(labels = ~`Energy Source`, values = ~`United States`, type =
      'pie', marker = list(line = list(width = 2)),hole = 0.7, sort = F
      ) %>%
  add_pie(data, labels = ~`Energy Source`, values = ~`European Union`,
        domain = list(
          x = c(0.15, 0.85),
```

```
261           y = c(0.15, 0.85)),
262         sort = F)
263
264 plot2 <- plot_ly(overview2) %>%
265   add_pie(overview2, labels = ~`Variants`, values = ~`count_variants`,
266         domain = list(
267           x = c(0.15, 0.85),
268           y = c(0.15, 0.85)),
269         type = 'pie', hole = 0,
270         sort = F)
271
272 plot4 <- plot_ly(overview1) %>%
273   add_pie(overview2, labels = ~`mutationType`, values = ~`count_
        variants`,
274         domain = list(
275           x = c(0.15, 0.85),
276           y = c(0.15, 0.85)),
277         type = 'pie', hole = 0,
278         sort = F)
279
280 ### PLOT 2a ### Quality of the variant data _____________________
      ____________________________________________________________
      ____________________________________________________________
281
282 sequencing_QA <- variants %>%
283   distinct(SOURCE, Patient_ID, Patient_NR, Master_ID, CHROM, POS, REF,
        ALT, .keep_all = T) %>%
284   mutate(total_reads=TumorAD+TumorRD)
285
286 #for labeling
287 sequencing_QA$SOURCE <- as_factor(sequencing_QA$SOURCE)
288 levels(sequencing_QA$SOURCE) <- c("RNA", "DNA")
289
290 ggplot(sequencing_QA, aes(x=reorder(SOURCE, desc(SOURCE)), y=total_
      reads, fill=SOURCE))+
291   geom_boxplot(varwidth = TRUE, outlier.alpha = 0.01, outlier.color = "
        grey", outlier.size = 5, alpha=0.7, lwd=1)+
292   stat_summary(fun = median, fun.max = length,
293             geom = "text", aes(label = paste("N=", ..ymax..)), size=10,
                  alpha=0.7, vjust = -1) +
294   scale_fill_manual(values=c("#bae4b3", "#2171b5"))+
295   #geom_hline(aes(yintercept=5), color="red", linetype="dotted", size
        =1.5)+
296   coord_cartesian(ylim = c(0, 500))+
297   theme_basic()+
298   theme(legend.position = "none")+
```

```
299   #facet_grid(~fct_infreq(Effect_group))+
300   labs(y="Coverage per altered locus (total reads)", x="", fill="")
301 
302 speichern("total_reads_boxplot_2")
303 
304 ggplot(sequencing_QA, aes(x=reorder(SOURCE, desc(SOURCE)), y=TumorVF,
        fill=SOURCE))+
305   geom_boxplot(varwidth = TRUE, outlier.alpha = 0.01, outlier.color = "
        grey", outlier.size = 5, alpha=0.7, lwd=1)+
306   stat_summary(fun = median, fun.max = length,
307             geom = "text", aes(label = paste("N=", ..ymax..)), size=10,
                  alpha=0.7, vjust = -1) +
308   scale_fill_manual(values=c("#bae4b3", "#2171b5"))+
309   #geom_hline(aes(yintercept=5), color="red", linetype="dotted", size
        =1.5)+
310   coord_cartesian(ylim = c(0, 1))+
311   theme_basic()+
312   theme(legend.position = "none")+
313   #facet_grid(~fct_infreq(Effect_group))+
314   labs(y="Tumor VF", x="", fill="")
315 
316 speichern("TumorVF_boxplot")
317 
318 addline_format <- function(x,...){
319   gsub('\\s','\n',x)
320 }
321 
322 ggplot(sequencing_QA, aes(x=fct_infreq(Biotype_group), y=total_reads,
        fill=SOURCE))+
323   geom_boxplot(varwidth = TRUE, outlier.alpha = 0.01, outlier.color = "
        grey", outlier.size = 5, alpha=0.7, lwd=1)+
324   #stat_summary(fun = median, fun.max = length,
325    # geom = "text", aes(label = paste("N=", ..ymax..)), size=10, alpha
         =0.7, vjust = -1) +
326   scale_fill_manual(values=c("#bae4b3", "#2171b5"))+
327   #geom_hline(aes(yintercept=5), color="red", linetype="dotted", size
        =1.5)+
328   coord_cartesian(ylim = c(0, 500))+
329   theme_basic()+
330   theme(axis.text.x = element_text(size=24))+
331   labs(y="Coverage per altered locus (total reads)", x="", fill="")+
332   scale_x_discrete(labels=function(x){sub("\\s", "\n", x)})
333 
334 speichern("total_reads_boxplot_detail9")
335 
336 
```

```
337 p <- ggplot(sequencing_QA, aes(x=total_reads, y=TumorVF, color=SOURCE))
        +
338   geom_point()+
339   scale_color_manual(values=c("#bae4b3", "#2171b5"))+
340   coord_cartesian(xlim = c(0, 2000))+
341   theme_basic()+
342   labs(x="Total reads", y="Tumor VF", color="")
343 
344 ggMarginal(p, type="boxplot")
345 
346 ggplot(sequencing_QA, aes(x=total_reads, fill=filter5))+
347   geom_density(alpha=0.5, adjust = 0.6)+
348   geom_vline(aes(xintercept=5), color="red", linetype="dotted", size
        =1.5)+
349   scale_fill_manual(values=c("#6495ED", "#FFD54D"))+
350   scale_y_continuous(breaks = scales::breaks_extended(n = 5))+
351   scale_x_log10(breaks=c(1,10,100,1000,10000,100000), limits=c
        (0.5,100000))+
352   theme_basic()+
353   theme(panel.grid.major.x=element_line())+
354   facet_grid(rows = vars(reorder(SOURCE, desc(SOURCE))))+
355   labs(x="Total reads", y="Density distribution", fill="Filtering")
356 
357 speichern("total_reads_density")
358 
359 
360 ggplot(sequencing_QA, aes(x=total_reads, fill=filter5))+
361   geom_histogram(binwidth = 0.1)+
362   geom_vline(aes(xintercept=5), color="red", linetype="dotted", size
        =1.5)+
363   scale_fill_manual(values=c("#6495ED", "#FFD54D"))+
364   scale_y_continuous(breaks = scales::breaks_extended(n = 5))+
365   scale_x_log10(breaks=c(1,10,100,1000,10000,100000), limits=c
        (0.5,100000))+
366   theme_basic()+
367   theme(panel.grid.major.x=element_line())+
368   facet_grid(rows = vars(reorder(SOURCE, desc(SOURCE))), scales="free_y
        ")+
369   labs(x="Total reads", y="Variants (count)", fill="Filtering")
370 
371 speichern("total_reads_histogram")
372 
373 
374 ggplot(sequencing_QA, aes(x=TumorVF, fill=filter5))+
375   geom_density(alpha=0.5, adjust = 0.1)+
376   geom_vline(aes(xintercept=0.05), color="red", linetype="dotted", size
```

```
      =1.5)+
  scale_fill_manual(values=c("#6495ED", "#FFD54D"))+
  scale_y_continuous(breaks = scales::breaks_extended(n = 5))+
  theme_basic()+
  theme(panel.grid.major.x=element_line())+
  facet_grid(rows = vars(reorder(SOURCE, desc(SOURCE))), scales="free_y
      ")+
  labs(x="Tumor VF", y="Density distribution", fill="Filtering")

speichern("TumorVF_density")


ggplot(sequencing_QA, aes(x=TumorVF, fill=filter5))+
  geom_histogram(bins = 100, col="grey")+
  geom_vline(aes(xintercept=0.05), color="red", linetype="dotted", size
      =1.5)+
  scale_fill_manual(values=c("#6495ED", "#FFD54D"))+
  scale_y_continuous(breaks = scales::breaks_extended(n = 5))+
  theme_basic()+
  theme(panel.grid.major.x=element_line())+
  facet_grid(rows = vars(reorder(SOURCE, desc(SOURCE))), scales="free_y
      ")+
  labs(x="Tumor VF", y="Variants (count)", fill="Filtering")

speichern("TumorVF_histogram")


### PLOT 2b ### Inliers vs. Outliers ____________________________________
    ____________________________________________________________________
    ____________________________________________________

Filtering <- variants_unique %>%
  group_by(filter5, SOURCE) %>%
  filter(!Pat_Met %in% c("11_T1","16_T1","20_T1","34_T1","31_T1","14_T1
      ","25_T1","25_T2")) %>%
  summarise(N.filter5=n()) %>%
  ungroup() %>%
  group_by(SOURCE) %>%
  mutate(perc = 100*round(N.filter5/colSums(across(where(is.numeric)))
      ,3)) %>%
  arrange(desc(N.filter5)) %>%
  arrange(match(filter5, c("Inliers", "Outliers"))) %>% #workaroung
      damit die Bennenung der labels die richtige Reihenfolge hat
  mutate(csum = rev(cumsum(rev(perc))),
         pos = perc/2 + lead(csum, 1),
         pos = if_else(is.na(pos), perc/2, pos))
```

```
Filtering$SOURCE <- as_factor(Filtering$SOURCE)
levels(Filtering$SOURCE) <- c("RNA", "DNA")
Filtering$annotation <- c("128.700", "8.800", NA, NA)
#Filtering$color <- c("#31a354", "#3182bd", "#e5f5e0", "#deebf7")

### PLOT
cp <- coord_polar(theta = "y")
cp$is_free <- function() TRUE
ggplot(Filtering, aes(x = 1, y = perc, fill = fct_inorder(filter5))) +
  geom_col(width = 0.5, color = 1) +
  geom_label_repel(data = Filtering,
                   aes(y = pos, label = paste0(N.filter5,"\n",perc, "%")),
                   size = 12, nudge_x = 1, show.legend = FALSE,
                   box.padding = 0.5) +
  geom_text(aes(x=0.2, y=0, label=annotation), size=14)+
  cp+
  #scale_fill_manual(values=c("#A7C7E7", "#6495ED", "#31a354", "#e5f5e0
      "))+
  scale_fill_brewer(palette = "Greens", direction=-1)+
  #scale_fill_brewer(palette = "Pastel1") +
  theme_void()+
  theme_donut()+
  theme(legend.position="bottom", legend.title = element_blank())+
  guides(fill = guide_legend(title = "Filtering")) +
  xlim(c(0.2, 1.5))+
  facet_wrap(~reorder(SOURCE, desc(SOURCE)), scales = "free")

speichern("Filtering_greens")


### PLOT 3 ### Variant frequencies (DNA and RNA) - filtered and
    unfiltered ____________________________________________________
    ____________________________________________________________
    ___________________________________

count_variants <- variants %>%
  distinct(SOURCE, Patient_ID, Patient_NR, Master_ID, CHROM, POS, REF,
      ALT, .keep_all = T) %>%
  group_by(SOURCE, filter5, Master_ID) %>%
  summarize(count_variants=n(), Pat_Met=first(Pat_Met), Patient_NR=
      first(Patient_NR), Patient_NR_plot=first(Patient_NR_plot),
      Metastasis=first(Metastasis), Metastasis_plot=first(Metastasis_
      plot), Tumor_entity=first(Tumor_entity), Tumor_entity_short_
      simple=first(Tumor_entity_short_simple), mean_TumorVF=mean(
      TumorVF)) %>%
  ungroup() %>% # Add empty Entries for Strelka manually (no RNA data
```

```
    for some Samples)
450 add_row(SOURCE="StrelkaRNA", filter5="Inliers", Master_ID="64EMZ9_T1"
    , count_variants=0, Patient_NR="11", Patient_NR_plot="11",
    Metastasis="T1", Metastasis_plot="T1", Tumor_entity="Endometrium-
    CA", Tumor_entity_short_simple="Carcinoma") %>%
451 add_row(SOURCE="StrelkaRNA", filter5="Outliers", Master_ID="64EMZ9_T1
    ", count_variants=0, Patient_NR="11", Patient_NR_plot="11",
    Metastasis="T1", Metastasis_plot="T1", Tumor_entity="Endometrium-
    CA", Tumor_entity_short_simple="Carcinoma") %>%
452 add_row(SOURCE="StrelkaRNA", filter5="Inliers", Master_ID="9YW2AD_T1"
    , count_variants=0, Patient_NR="16", Patient_NR_plot="16",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="Adenocarcinoma
    ", Tumor_entity_short_simple="Carcinoma") %>%
453 add_row(SOURCE="StrelkaRNA", filter5="Outliers", Master_ID="9YW2AD_T1
    ", count_variants=0, Patient_NR="16", Patient_NR_plot="16",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="Adenocarcinoma
    ", Tumor_entity_short_simple="Carcinoma") %>%
454 add_row(SOURCE="StrelkaRNA", filter5="Inliers", Master_ID="M218BR_T1"
    , count_variants=0, Patient_NR="20", Patient_NR_plot="20",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="Testicle-CA",
    Tumor_entity_short_simple="Carcinoma") %>%
455 add_row(SOURCE="StrelkaRNA", filter5="Outliers", Master_ID="M218BR_T1
    ", count_variants=0, Patient_NR="20", Patient_NR_plot="20",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="Testicle-CA",
    Tumor_entity_short_simple="Carcinoma") %>%
456 add_row(SOURCE="StrelkaRNA", filter5="Inliers", Master_ID="LRE6DV_T1"
    , count_variants=0, Patient_NR="34", Patient_NR_plot="34",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="Mucinous
    Adenocarcinoma", Tumor_entity_short_simple="Carcinoma") %>%
457 add_row(SOURCE="StrelkaRNA", filter5="Outliers", Master_ID="LRE6DV_T1
    ", count_variants=0, Patient_NR="34", Patient_NR_plot="34",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="Mucinous
    Adenocarcinoma", Tumor_entity_short_simple="Carcinoma") %>%
458 add_row(SOURCE="StrelkaRNA", filter5="Inliers", Master_ID="NLSTH8_T1"
    , count_variants=0, Patient_NR="31", Patient_NR_plot="31",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="
    Rhabdomyosarcoma", Tumor_entity_short_simple="Sarcoma") %>%
459 add_row(SOURCE="StrelkaRNA", filter5="Outliers", Master_ID="NLSTH8_T1
    ", count_variants=0, Patient_NR="31", Patient_NR_plot="31",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="
    Rhabdomyosarcoma", Tumor_entity_short_simple="Sarcoma") %>%
460 add_row(SOURCE="StrelkaRNA", filter5="Inliers", Master_ID="XVM4XC_T1"
    , count_variants=0, Patient_NR="14", Patient_NR_plot="14",
    Metastasis="T1", Metastasis_plot="", Tumor_entity="Melanoma",
    Tumor_entity_short_simple="Melanoma") %>%
461 add_row(SOURCE="StrelkaRNA", filter5="Outliers", Master_ID="XVM4XC_T1
```

```
      ", count_variants=0, Patient_NR="14", Patient_NR_plot="14",
      Metastasis="T1", Metastasis_plot="", Tumor_entity="Melanoma",
      Tumor_entity_short_simple="Melanoma") %>%
462  add_row(SOURCE="StrelkaRNA", filter5="Inliers", Master_ID="42D9U7_T1"
      , count_variants=0, Patient_NR="25", Patient_NR_plot="25",
      Metastasis="T1", Metastasis_plot="T1", Tumor_entity="WT-GIST",
      Tumor_entity_short_simple="other") %>%
463  add_row(SOURCE="StrelkaRNA", filter5="Outliers", Master_ID="42D9U7_T1
      ", count_variants=0, Patient_NR="25", Patient_NR_plot="25",
      Metastasis="T1", Metastasis_plot="T1", Tumor_entity="WT-GIST",
      Tumor_entity_short_simple="other") %>%
464  add_row(SOURCE="StrelkaRNA", filter5="Inliers", Master_ID="42D9U7_T2"
      , count_variants=0, Patient_NR="25", Patient_NR_plot=NA,
      Metastasis="T2", Metastasis_plot="T2", Tumor_entity="WT-GIST",
      Tumor_entity_short_simple="other") %>%
465  add_row(SOURCE="StrelkaRNA", filter5="Outliers", Master_ID="42D9U7_T2
      ", count_variants=0, Patient_NR="25", Patient_NR_plot=NA,
      Metastasis="T2", Metastasis_plot="T2", Tumor_entity="WT-GIST",
      Tumor_entity_short_simple="other") %>%
466  mutate(Tumor_entity_short_simple = fct_reorder(Tumor_entity_short_
      simple, count_variants, .fun='length' ))
467
468 count_variants_DNA <- count_variants %>%
469  filter(SOURCE=="Mutect2")
470 count_variants_RNA <- count_variants %>%
471  filter(SOURCE=="StrelkaRNA")
472
473 p1 <- ggplot(count_variants_DNA, aes(x=reorder(Pat_Met, desc(Tumor_
     entity_short_simple)), y=count_variants, fill=reorder(filter5, desc
     (filter5)), width=.75))+
474  geom_bar(colour="black", stat = "identity")+
475  scale_y_continuous(breaks = scales::breaks_extended(n = 6))+
476  #theme_minimal()+
477  theme_basic()+
478  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank(),
      panel.grid.major.x=element_blank())+
479  geom_label(aes(label=Patient_NR_plot, y=max(count_variants)/-11),
      fill="white", size = 8, label.r=unit(0.5,"lines"), label.padding=
      unit(0.08,"lines"))+
480  geom_text(aes(label=Metastasis_plot, y=max(count_variants)/-33), size
      =8)+
481  geom_hline(yintercept = 0)+
482  scale_fill_brewer(palette = "Blues")+
483  #scale_fill_manual(values=c("#A7C7E7", "#6495ED"))+
484  labs(x="", y="DNA variants (count)", fill="")
485
```

```
p2 <- ggplot(count_variants_RNA, aes(x=reorder(Pat_Met, desc(Tumor_entity_short_simple)), y=count_variants, fill=reorder(filter5, desc(filter5)), width=.75))+
  geom_bar(colour="black", stat = "identity")+
  scale_y_continuous(breaks = scales::breaks_extended(n = 6))+
  #theme_minimal()+
  theme_basic()+
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank(), panel.grid.major.x=element_blank())+
  geom_label(aes(label=Patient_NR_plot, y=max(count_variants)/-9), fill="white", size = 8, label.r=unit(0.5,"lines"), label.padding=unit(0.08,"lines"))+
  geom_text(aes(label=Metastasis_plot, y=max(count_variants)/-30), size=8)+
  geom_hline(yintercept = 0)+
  scale_fill_brewer(palette = "Greens")+
  #scale_fill_manual(values=c("#A7C7E7", "#6495ED"))+
  labs(x="Patient ID, Entity, Tumor", y="RNA variants (count)", fill="")

grid.newpage()
g <- grid.arrange(p1, p2)
speichern_grid("VariantsCount_test_3")


### PLOT 4 ### DNA/RNA overlap (all patients and subgroup of patients) ----------------------------------------------------------------------------------------------------------------------

coverage_all <- variants %>%
  distinct(SOURCE, Patient_ID, Patient_NR, Master_ID, CHROM, POS, REF, ALT, .keep_all = T) %>%
  filter(!Pat_Met %in% c("11_T1","16_T1","20_T1","34_T1","31_T1","14_T1","25_T1","25_T2")) %>%
  #filter(filter5=="Inliers") %>%
  #filter(Tumor_entity_short_simple=="other") %>%
  mutate(variant=paste0(Master_ID, CHROM, POS, REF, ALT)) %>%
  group_by(SOURCE) %>%
  summarise(mutation=paste0(variant, collapse = ";"), N.mutations=n())
Mutect2 <- as.vector(str_split(coverage_all$mutation[1],";", simplify = T))
StrelkaRNA <- as.vector(str_split(coverage_all$mutation[2],";", simplify = T))
mutation.called.by <- list("WES"=Mutect2,"RNA-Seq"=StrelkaRNA)

```

```
Venn.plot.2(mutation.called.by, save.plot = T, "coverage_other_Inliers")

### Plot 5a ### :: Variant Type analysis :: ____________________________
_____________________________________________________________________
_____________________________________________________________

VT <- variants_unique %>%
  #filter(!Pat_Met %in% c("11_T1","16_T1","20_T1","34_T1","31_T1","14_T1","25_T1","25_T2")) %>%
  filter(filter5=="Inliers") %>%
  group_by(Effect_group, SOURCE) %>%
  summarise(N.effect_group=n()) %>%
  ungroup() %>%
  group_by(SOURCE) %>%
  mutate(perc = 100*round(N.effect_group/colSums(across(where(is.numeric))),3)) %>%
  arrange(desc(N.effect_group)) %>%
  arrange(match(Effect_group, c("Non-coding", "Coding missense", "Splice site & intron", "Stop gained", "Frameshift", "Others"))) %>% #workaroung for labeling to be in the right order
  mutate(csum = rev(cumsum(rev(perc))),
         pos = perc/2 + lead(csum, 1),
         pos = if_else(is.na(pos), perc/2, pos))
VT$SOURCE <- as_factor(VT$SOURCE)
levels(VT$SOURCE) <- c("RNA", "DNA")
VT <- VT %>%
  group_by(SOURCE) %>%
  mutate(Summen=sum(N.effect_group))
#VT$annotation <- c("128.700", "8.800", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA)
VT$annotation <- c("83.993", "4.631", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA)

### PLOT
cp <- coord_polar(theta = "y")
cp$is_free <- function() TRUE
ggplot(VT, aes(x = 1, y = perc, fill = fct_inorder(Effect_group))) +
  geom_col(width = 0.5, color = 1) +
  geom_label_repel(data = VT,
                   aes(y = pos, label = paste0(N.effect_group,"\n",perc, "%")),
                   size = 12, nudge_x = 1, show.legend = FALSE,
                   box.padding = 0.5) +
  geom_text(aes(x=0.2, y=0, label=annotation), size=14)+
  cp+
```

```
  scale_fill_brewer(palette = "Pastel1") +
  theme_void()+
  theme_donut()+
  guides(fill = guide_legend(title = "Variant type")) +
  xlim(c(0.2, 1.5))+
  facet_wrap(~reorder(SOURCE, desc(SOURCE)), scales = "free")

speichern("Variant_type_2")


### Plot 5b ### :: Mutation Type analysis :: ____________________________
    ____________________________________________________________________
    ____________________________________________________________________

MT <- variants_unique %>%
  #filter(!Pat_Met %in% c("11_T1","16_T1","20_T1","34_T1","31_T1","14_
      T1","25_T1","25_T2")) %>%
  filter(filter5=="Inliers") %>%
  group_by(mutationType, SOURCE) %>%
  summarise(N.mutationType=n()) %>%
  ungroup() %>%
  group_by(SOURCE) %>%
  mutate(perc = 100*round(N.mutationType/colSums(across(where(is.
      numeric))),3)) %>%
  arrange(desc(N.mutationType)) %>%
  arrange(match(mutationType, c("substitution", "deletion", "insertion"
      , "multi-substitution"))) %>% #workaroung damit die Bennenung der
       labels die richtige Reihenfolge hat
  mutate(csum = rev(cumsum(rev(perc))),
       pos = perc/2 + lead(csum, 1),
       pos = if_else(is.na(pos), perc/2, pos))
MT$SOURCE <- as_factor(MT$SOURCE)
levels(MT$SOURCE) <- c("RNA", "DNA")
MT <- MT %>%
  group_by(SOURCE) %>%
  mutate(Summen=sum(N.mutationType))
#MT$annotation <- c("128.700", "8.800", NA, NA, NA, NA, NA, NA)
MT$annotation <- c("83.993", "4.631", NA, NA, NA, NA, NA, NA)

### PLOT
cp <- coord_polar(theta = "y")
cp$is_free <- function() TRUE
ggplot(MT, aes(x = 1, y = perc, fill = fct_inorder(mutationType))) +
  geom_col(width = 0.5, color = 1) +
  geom_label_repel(data = MT,
                aes(y = pos, label = paste0(N.mutationType,"\n",perc, "%
```

```
                    ")),
594               size = 12, nudge_x = 1, show.legend = FALSE,
595               box.padding = 0.5) +
596  geom_text(aes(x=0.2, y=0, label=annotation), size=14)+
597  cp+
598  scale_fill_brewer(palette = "Pastel1") +
599  theme_void()+
600  theme_donut()+
601  guides(fill = guide_legend(title = "Mutation type")) +
602  xlim(c(0.2, 1.5))+
603  facet_wrap(~reorder(SOURCE, desc(SOURCE)), scales = "free")
604
605 speichern("Mutation_type_inliers_2")
606
607 MT <- variants_unique %>%
608  group_by(Tumor_ID, Mutation_ID) %>%
609  summarise(Effect_group=first(Effect_group), SOURCE=paste0(sort(unique
       (SOURCE)), collapse = "+")) %>%
610  ungroup() %>%
611  group_by(Effect_group, SOURCE) %>%
612  summarise(N.effect_group=n()) %>%
613  arrange(desc(N.effect_group))
614
615 ggplot(MT, aes(x=reorder(Effect_group, desc(N.effect_group)), y=N.
     effect_group, fill=SOURCE))+
616  geom_col(position = "dodge")
617
618
619 ### Plot 5c ### :: BioType analysis :: ____________________________
     ____________________________________________________________________
     ____________________________________________________________
620
621 BT <- variants_unique %>%
622  filter(filter5=="Inliers") %>%
623  group_by(Biotype_group, SOURCE) %>%
624  summarise(N.Biotype_group=n()) %>%
625  ungroup() %>%
626  group_by(SOURCE) %>%
627  mutate(perc = 100*round(N.Biotype_group/colSums(across(where(is.
       numeric))),3)) %>%
628  arrange(desc(N.Biotype_group)) %>%
629  arrange(match(Biotype_group, c("Protein Coding", "Regulatory RNAs", "
       Pseudogene", "Processed Transcript", "TEC", "Sense Intronic", "
       Others"))) %>% #workaroung damit die Bennenung der labels die
       richtige Reihenfolge hat
630  mutate(csum = rev(cumsum(rev(perc))),
```

```
631       pos = perc/2 + lead(csum, 1),
632       pos = if_else(is.na(pos), perc/2, pos))
633 BT$SOURCE <- as_factor(BT$SOURCE)
634 levels(BT$SOURCE) <- c("RNA", "DNA")
635 BT <- BT %>%
636  group_by(SOURCE) %>%
637  mutate(Summen=sum(N.Biotype_group))
638 #BT$annotation <- c("128.700", NA, NA, "9.900", NA, NA, NA, NA, NA, NA,
     NA, NA, NA, NA)
639 BT$annotation <- c("83.993", NA, NA, "4.631", NA, NA, NA, NA, NA, NA,
    NA, NA, NA, NA)
640
641 ### PLOT
642 cp <- coord_polar(theta = "y")
643 cp$is_free <- function() TRUE
644 ggplot(BT, aes(x = 1, y = perc, fill = fct_inorder(Biotype_group))) +
645  geom_col(width = 0.5, color = 1) +
646  geom_label_repel(data = BT,
647              aes(y = pos, label = paste0(N.Biotype_group,"\n",perc, "
                 %")),
648              size = 12, nudge_x = 1, show.legend = FALSE,
649              box.padding = 0.5) +
650  geom_text(aes(x=0.2, y=0, label=annotation), size=14)+
651  cp+
652  scale_fill_brewer(palette = "Pastel1") +
653  theme_void()+
654  theme_donut()+
655  guides(fill = guide_legend(title = "Genetic biotype")) +
656  xlim(c(0.2, 1.5))+
657  facet_wrap(~reorder(SOURCE, desc(SOURCE)), scales = "free")
658
659 speichern("Biotype_4")
660
661
662 ### Plot X ### :: Gene Analysis :: ____________________________________
    ____________________________________________________________________
    ____________________________________________________
663 genelength <- reference.genelength2[!duplicated(reference.genelength2),
     ] %>%
664  mutate(gene.length=end-start) %>%
665  distinct(transcriptID, .keep_all = TRUE) %>%
666  mutate(geneName.old = geneName, geneName=str_sub(geneName, 1, 8))
667
668 GA.badmatch <- variants %>%
669  select(Patient_ID, Patient_NR, FEATUREID, CHROM, Tumor_ID, Metastasis
      , Pat_Met, Mutation_ID, SOURCE, GENE, filter5, Tumor_entity,
```

```
      Tumor_entity_short, Tumor_entity_short_simple) %>%
  mutate(transcript.id=str_sub(FEATUREID, 1,15)) %>%
  #distinct(Pat_Met, Mutation_ID, SOURCE, filter5, .keep_all = TRUE)
      %>%
  distinct(Patient_ID, Mutation_ID, SOURCE, filter5, .keep_all = TRUE)
      %>%
  merge(genelength, by.x = "transcript.id", by.y = "transcriptID", all.
      x = TRUE) %>%
  filter(is.na(gene.length)) %>%
  mutate(GENE.merge=str_sub(GENE, 1,8)) %>%
  merge(distinct(genelength, geneName, .keep_all = TRUE), by.x = "GENE.
      merge", by.y = "geneName", all.x = TRUE, suffixes = c(".x", ""))
      %>%
  select(-contains(".x"), -GENE.merge, -geneName)

GA <- variants %>%
  select(Patient_ID, Patient_NR, FEATUREID, CHROM, Tumor_ID, Metastasis
      , Pat_Met, Mutation_ID, SOURCE, GENE, filter5, Tumor_entity,
      Tumor_entity_short, Tumor_entity_short_simple) %>%
  mutate(transcript.id=str_sub(FEATUREID, 1,15)) %>%
  #distinct(Pat_Met, Mutation_ID, SOURCE, filter5, .keep_all = TRUE)
      %>%
  distinct(Patient_ID, Mutation_ID, SOURCE, filter5, .keep_all = TRUE)
      %>%
  merge(genelength, by.x = "transcript.id", by.y = "transcriptID", all.
      x = TRUE) %>%
  filter(!is.na(gene.length)) %>%
  bind_rows(GA.badmatch) %>%
  select(-geneName) %>%
  rename(geneName="geneName.old") %>% #GENE: hat keine NAs, geneName:
      HAT 114 NA-Eintraege, wo keine Zuordnung und damit keine
      genelength bestimmt werden konnte
  # group and filter such, that multi metastasis sharing variants are
      only counted ONCE.
  mutate(temp=ifelse(filter5=="Inliers", 2, 1)) %>%
  group_by(Patient_ID, Mutation_ID) %>%
  filter(temp==max(temp)) %>%
  select(-temp) %>%
  group_by(GENE, SOURCE) %>%
  mutate(N.variants.per.gen.test=n()) %>%
  ungroup()

GA.filter <- GA %>%
  group_by(GENE, Pat_Met, SOURCE, filter5) %>%
  summarise(N.variants.per.gene.per.tumor=n(), N.variants.per.gen.test=
      first(N.variants.per.gen.test), geneName=paste(unique(geneName),
```

```
      collapse = "; "), Tumor_entity=paste(unique(Tumor_entity),
      collapse = " + "), Tumor_entity_short=unique(Tumor_entity_short),
       Tumor_entity_short_simple=paste(unique(Tumor_entity_short_simple
      ), collapse = " + "), gene.length=mean(gene.length)) %>%
  #mutate(N.variants.normalized=ifelse(is.na(gene.length), NA, 100*N.
      variants.per.gen.test/gene.length)) %>%
  group_by(GENE, SOURCE, filter5) %>%
  summarise(N.variants.per.gene=sum(N.variants.per.gene.per.tumor),
      geneName=paste(unique(geneName), collapse = "; "), Tumor_entity_
      short_simple=paste(unique(Tumor_entity_short_simple), collapse =
      " + "), gene.length=mean(gene.length)) %>%
  mutate(N.variants.normalized=ifelse(is.na(gene.length), NA, 100*N.
      variants.per.gene/gene.length)) %>%
  arrange(desc(N.variants.normalized)) %>%
  group_by(GENE, SOURCE) %>%
  mutate(N.variants.normalized.sum=sum(N.variants.normalized)) %>%
  mutate(filter5=as.factor(filter5)) %>%
  filter(SOURCE=="Mutect2") %>%
  filter(N.variants.normalized.sum>0.41)
  #filter(N.variants.normalized.sum>8.3)

ggplot(GA.filter, aes(x=reorder(GENE, N.variants.normalized.sum), y=N.
    variants.normalized, fill=fct_rev(filter5)))+
  geom_col(position = "stack")+
  coord_flip()+
  theme_minimal()+
  theme(legend.key.size = unit(2.5, "cm"), legend.text = element_text(
      size=30), legend.title = element_text(size=30))+
  theme(axis.title = element_text(size=30), axis.text = element_text(
      size=30), axis.text.y = element_text(size=26), axis.text.y.right
      = element_blank(), axis.ticks.x=element_blank())+
  theme(panel.grid.major.y = element_blank(), panel.grid.minor=element_
      blank(), strip.text.y = element_text(size = 30))+
  scale_fill_manual(values=c(alpha("#6495ED", 0.8), alpha("#FFD54D",
      0.8)), breaks=c("Inliers", "Outliers"), labels=c("Filtered", "
      Outliers"))+
  labs(x="Gene", y="Mutations per 100bp", fill="")

speichern("GA.1_1_11i")


 GA.entity <- GA %>%
  group_by(GENE, SOURCE, filter5, Tumor_entity_short_simple) %>%
  summarise(N.variants.per.gene.per.tumor=n(), N.variants.per.gen.test=
      first(N.variants.per.gen.test), geneName=paste(unique(geneName),
      collapse = "; "), Tumor_entity=paste(unique(Tumor_entity),
```

```
      collapse = " + "), gene.length=mean(gene.length)) %>%
729  mutate(N.variants.normalized=ifelse(is.na(gene.length), NA, 100*N.
      variants.per.gene.per.tumor/gene.length)) %>%
730  group_by(GENE, SOURCE, filter5) %>%
731  mutate(N.variants.normalized.sum=sum(N.variants.normalized)) %>%
732  select(-N.variants.per.gen.test) %>%
733  filter(filter5=="Inliers") %>%
734  mutate(Tumor_entity_short_simple=as.factor(Tumor_entity_short_simple)
      ) %>%
735  group_by(GENE, SOURCE) %>%
736  mutate(N.variants.normalized.total=sum(N.variants.normalized), N.
      variants.per.gene.per.tumor.sum=sum(N.variants.per.gene.per.tumor
      )) %>%
737  arrange(desc(N.variants.normalized.total)) %>%
738  filter(SOURCE=="Mutect2") %>%
739  filter(N.variants.per.gene.per.tumor.sum>2) %>%
740  #filter(N.variants.normalized.total>8.3)
741  filter(N.variants.normalized.total>0.02)
742
743 # scheme plot RNA
744 ggplot(GA.entity, aes(x=fct_rev(fct_inorder(GENE)), y=N.variants.
     normalized, fill=factor(Tumor_entity_short_simple, levels = c("
     Other", "Melanoma", "Sarcoma", "Carcinoma"))))+
745  geom_col(position = "stack")+
746  theme_minimal()+
747  theme(legend.key.size = unit(2.5, "cm"), legend.text = element_text(
      size=30), legend.title = element_text(size=30))+
748  theme(axis.title = element_text(size=30), axis.text = element_text(
      size=30), axis.text.y = element_text(size=30), axis.text.y.right
      = element_blank(), axis.ticks.x=element_blank())+
749  theme(panel.grid.major.y = element_blank(), panel.grid.minor=element_
      blank(), strip.text.y = element_text(size = 30))+
750  coord_flip()+
751  scale_fill_brewer(palette="Pastel1", breaks=c("Carcinoma", "Sarcoma",
       "Melanoma", "Other"), labels=c("Carcinoma", "Sarcoma", "Melanoma
      ", "Other"))+
752  labs(x="Gene", y="Mutations per 100bp", fill="")
753
754 # scheme plot DNA
755 ggplot(GA.entity, aes(x=fct_rev(fct_inorder(GENE)), y=N.variants.
     normalized, fill=gene.length))+
756  geom_col(position = "stack")+
757  theme_minimal()+
758  theme(legend.key.size = unit(2.5, "cm"), legend.text = element_text(
      size=30), legend.title = element_text(size=30))+
759  theme(axis.title = element_text(size=30), axis.text = element_text(
```

```
      size=30), axis.text.y = element_text(size=26), axis.text.x.top =
      element_blank(), axis.ticks.x=element_blank())+
  theme(panel.grid.major.y = element_blank(), panel.grid.minor=element_
      blank(), strip.text.y = element_text(size = 30))+
  coord_flip()+
  scale_y_continuous(limits=c(0,2.4), breaks = c(0, 0.05, 0.1, 0.15,
      0.20, 0.25, 0.3, 2.2, 2.3, 2.4))+
  scale_y_break(c(0.26,2.3), scales = 0.12, space = 0.8)+
  #scale_fill_brewer(palette="Pastel1", breaks=c("Carcinoma", "Sarcoma
      ", "Melanoma", "Other"), labels=c("Carcinoma", "Sarcoma", "
      Melanoma", "Other"))+
  scale_fill_viridis_c(direction = -1, alpha = 0.6, breaks=c
      (0,5000,10000,15000,20000,25000,30000,35000), limits=c(0,36000),
      begin = 0, end = 0.95)+
  labs(x="Gene", y="Mutations per 100bp", fill="Genesize \n[bp]")

# Detail plots
GA.help <- GA %>%
  distinct(Patient_ID, Tumor_entity_short_simple) %>%
  group_by(Tumor_entity_short_simple) %>%
  summarise(N.patients.per.entity=n())

GA.entity.single <- GA.entity %>%
  mutate(GENE=as.factor(GENE)) %>%
  arrange(desc(N.variants.normalized)) %>%
  group_by(Tumor_entity_short_simple) %>%
  mutate(row.rank=row_number()) %>%
  ungroup() %>%
  filter(row.rank<16) %>%
  merge(GA.help, by.x = "Tumor_entity_short_simple", by.y = "Tumor_
      entity_short_simple") %>%
  # for DNA: *1000 for better readability
  #mutate(N.variants.normalized.per.patient=N.variants.normalized/N.
      patients.per.entity)
  mutate(N.variants.normalized.per.patient=1000*N.variants.normalized/N
      .patients.per.entity)
expression(paste("Volume ", m^{3}))
# RNA
for (i in c("Carcinoma", "Sarcoma", "Melanoma", "Other")) {
  temp <- ggplot(filter(GA.entity.single, Tumor_entity_short_simple==i)
      , aes(x=reorder(GENE, N.variants.normalized), y=N.variants.
      normalized.per.patient, fill=gene.length))+
    geom_col()+
    theme_basic( )+
    coord_flip()+
    scale_fill_viridis_c(direction = -1, alpha = 0.6, breaks=c
```

```
        (0,500,1000,1500,2000,2500), limits=c(0,2500), begin = 0, end =
        1)+
793    labs(x=ifelse(i %in% c("Carcinoma", "Melanoma"), "Gene", ""), y=
        ifelse(i %in% c("Melanoma", "Other"), "Mutations per 100bp (mean
        )", ""), fill="Genesize \n[bp]")+
794    #guides(fill = guide_legend(override.aes = list(size = 1)))
795    theme_minimal()+
796    theme(legend.key.size = unit(2, "cm"), legend.key.width = unit(1, "
        cm"), legend.text = element_text(size=26), legend.title =
        element_text(size=26))+
797    theme(axis.title = element_text(size=26), axis.text = element_text(
        size=26), axis.text.y = element_text(size=22), axis.text.y.right
         = element_blank(), axis.ticks.x=element_blank())+
798    theme(panel.grid.major.y = element_blank(), panel.grid.minor=element
        _blank(), strip.text.y = element_text(size = 26))+
799    #annotate("text", x = 5.0, y = ifelse(i=="Carcinoma", 9 ,ifelse(i=="
        Sarcoma", 7.2 , ifelse(i=="Melanoma", 4 , 0.75 ))),
800     # label = ifelse(i=="Carcinoma","Carcinoma" ,ifelse(i=="Sarcoma", "
         Sarcoma" , ifelse(i=="Melanoma","Melanoma" ,"Other"))),
801      # angle=90, size=12, vjust = 0, alpha=0.3)
802    annotate("text", x = 7.5, y = ifelse(i=="Carcinoma", 0.53 ,ifelse(i
        =="Sarcoma", 0.7 , ifelse(i=="Melanoma", 0.85 , 0.34 ))),
803           label = ifelse(i=="Carcinoma","Carcinoma" ,ifelse(i=="Sarcoma"
                , "Sarcoma" , ifelse(i=="Melanoma","Melanoma" ,"Other"))),
804          angle=90, size=12, vjust = 0, alpha=0.4)
805  nam=paste0("plot_",i)
806  assign(nam, temp)
807 }
808
809 ggarrange(plot_Carcinoma, plot_Sarcoma, plot_Melanoma, plot_Other, ncol
    =2, nrow=2, common.legend = TRUE, legend="right")
810
811 speichern("grid_7_b")
812
813
814 # DNA
815 for (i in c("Carcinoma", "Sarcoma", "Melanoma", "Other")) {
816  temp <- ggplot(filter(GA.entity.single, Tumor_entity_short_simple==i)
      , aes(x=reorder(GENE, N.variants.normalized), y=N.variants.
      normalized.per.patient, fill=gene.length))+
817    geom_col()+
818    theme_basic( )+
819    coord_flip()+
820    #scale_fill_viridis_c(direction = -1, alpha = 0.6, breaks=c
        (0,5000,10000,15000,20000,25000,30000,35000), limits=c(0,36000),
         begin = 0, end = 0.95, na.value = "black")+
```

```
821    scale_fill_viridis_c(direction = -1, alpha = 0.6, breaks=c
           (0,10000,20000,30000,40000,50000), limits=c(0,50000), begin = 0,
            end = 0.95, na.value = "black")+
822    labs(x=ifelse(i %in% c("Carcinoma", "Melanoma"), "Gene", ""), y=
           ifelse(i %in% c("Melanoma", "Other"), expression(paste("
           Mutations per 100bp (mean) [ x ", 10^{-3}, " ]")), ""), fill="
           Genesize \n[bp]")+
823    #guides(fill = guide_legend(override.aes = list(size = 1)))
824    theme_minimal()+
825    theme(legend.key.size = unit(2, "cm"), legend.key.width = unit(1, "
           cm"), legend.text = element_text(size=26), legend.title =
           element_text(size=26))+
826    theme(axis.title = element_text(size=26), axis.text = element_text(
           size=26), axis.text.y = element_text(size=22), axis.text.y.right
            = element_blank(), axis.ticks.x=element_blank())+
827    theme(panel.grid.major.y = element_blank(), panel.grid.minor=element
           _blank(), strip.text.y = element_text(size = 10))
828    #scale_y_continuous(limits=ifelse(i=="Carcinoma", c(0,0.055) ,ifelse
           (i=="Sarcoma", c(0,0.16) , ifelse(i=="Melanoma", c(0,0.017) , c
           (0,0.006) ))),
829    # breaks = ifelse(i=="Carcinoma", c(0,0.05) ,ifelse(i=="Sarcoma", c
           (0,0.15) , ifelse(i=="Melanoma", c(0,0.015) , c(0,0.002, 0.004)
           ))))#+
830    #annotate("text", x = 7.5, y = ifelse(i=="Carcinoma", 0.53 ,ifelse(i
           =="Sarcoma", 0.7 , ifelse(i=="Melanoma", 0.85 , 0.34 ))),
831     # label = ifelse(i=="Carcinoma","Carcinoma" ,ifelse(i=="Sarcoma", "
            Sarcoma" , ifelse(i=="Melanoma","Melanoma" ,"Other"))),
832      # angle=90, size=12, vjust = 0, alpha=0.4)
833  nam=paste0("plot_",i)
834  assign(nam, temp)
835 }
836
837 plot_Carcinoma <- plot_Carcinoma+
838 #plot_Carcinoma+
839  scale_y_continuous(limits=c(0,60), breaks = c(0,1,2,3,4,5,50,60)) +
840  scale_y_break(c(5,48), scales = 0.12, space = 0.6)+
841  theme(axis.text.x.top = element_blank())+
842  annotate("text", x = 7.5, y = 4.6 , label = "Carcinoma", angle=90,
         size=12, vjust = 0, alpha=0.4)+
843  guides(fill="none")
844
845
846 plot_Sarcoma <- plot_Sarcoma+
847 #plot_Sarcoma+
848  scale_y_continuous(limits=c(0,165), breaks = c
         (0,5,10,15,20,25,150,160)) +
```

```
849   scale_y_break(c(25,145), scales = 0.30, space = 0.6)+
850   theme(axis.text.x.top = element_blank())+
851   annotate("text", x = 7.5, y = 23 , label = "Sarcoma", angle=90, size
          =12, vjust = 0, alpha=0.4)+
852   guides(fill="none")
853
854 plot_Melanoma <- plot_Melanoma+
855 #plot_Melanoma+
856   scale_y_continuous(limits=c(0,20), breaks = c(0,2,4,6,8,15,20)) +
857   scale_y_break(c(8,15), scales = 0.12, space = 0.6)+
858   theme(axis.text.x.top = element_blank())+
859   annotate("text", x = 7.5, y = 7.25 , label = "Melanoma", angle=90,
          size=12, vjust = 0, alpha=0.4)+
860   guides(fill="none")
861
862 plot_Other <- plot_Other+
863   annotate("text", x = 3.5, y = 3.7 , label = "Other", angle=90, size
          =12, vjust = 0, alpha=0.4)
864
865
866 ggarrange(print(plot_Carcinoma), print(plot_Sarcoma), print(plot_
      Melanoma), print(plot_Other), ncol=2, nrow=2, common.legend = TRUE,
       legend="right")
867 speichern("grid_8_j")
868    #speichern_grid("grid_2")
869
870 #
      ##############################################################################
871
872 ggplot(GA.entity.single, aes(x=reorder(GENE, N.variants.normalized.
      total), y=N.variants.normalized, fill=gene.length))+
873   geom_col()+
874   theme_basic()+
875   coord_flip()+
876   #scale_fill_brewer(palette="Pastel1", breaks=c("Carcinoma", "Sarcoma
          ", "Melanoma", "Other"), labels=c("Carcinoma", "Sarcoma", "
          Melanoma", "Other"))+
877   scale_fill_viridis_c(direction = -1, alpha = 0.6, breaks=c
          (0,500,1000,1500,2000,2500), limits=c(0,2500))+
878   labs(x="Gene", y="mutations per 100bp", fill="")+
879   facet_grid2(rows = vars(factor(Tumor_entity_short_simple, levels = c(
          "Carcinoma", "Sarcoma", "Melanoma", "Other"))), scales = "free",
          independent="all")
880
881
```

```
882 ### Plot 6 ### :: Shared Mutations :: ____________________________________
    ____________________________________________________________________________
    ____________________________________________________________
883
884 ### DNA ###
885 SM_DNA <- variants_unique %>%
886   filter(SOURCE=="Mutect2") %>%
887   group_by(Patient_ID, Mutation_ID) %>%
888   summarise(N.tumor.per.mut.per.patient=n(), GENE=paste0(unique(GENE),
        collapse = ";"), Metastasis=paste0(unique(Metastasis), collapse="
        ,"), mutationType=paste0(unique(mutationType), collapse = ";"),
        Effect_group=paste0(unique(Effect_group), collapse = ";"), Tumor_
        entity=paste0(unique(Tumor_entity), collapse = ";")) %>%
889   ungroup() %>% group_by(Mutation_ID) %>%
890   summarise(N.patient.per.mut=n(), N.total.per.mut=sum(N.tumor.per.mut.
        per.patient), GENE=paste0(unique(as.vector(str_split(paste0(GENE,
         collapse = ";"),";", simplify = T))),collapse=";"), mutationType
        =paste0(unique(mutationType), collapse = ";"), Effect_group=
        paste0(unique(Effect_group), collapse = ";"), Tumor_entity=paste0
        (unique(as.vector(str_split(paste0(Tumor_entity, collapse = ";"),
        ";", simplify = T))),collapse=";"))
891   #summarise(N.patient.per.mut=n(), N.total.per.mut=sum(N.tumor.per.mut
        .per.patient), GENE=paste(unique(GENE), collapse = "+++"))
892
893 ggplot(SM_DNA %>% slice_max(N.patient.per.mut, n=15), aes(width=.75))+
894   geom_col(aes(x=reorder(Mutation_ID, desc(-N.patient.per.mut)), y=N.
        patient.per.mut), fill=alpha("#2171b5", 0.4), color="black")+
895   geom_label(size=8, aes(x=reorder(Mutation_ID, desc(-N.patient.per.mut
        )), y= 0.5, hjust = 0, label=GENE))+
896   scale_y_continuous(breaks = scales::breaks_extended(n = 8))+
897   coord_flip()+
898   labs(y="Patients carrying the mutation (count)", x="Mutation ID (with
         corresponding Gene)")+
899   theme_basic()+
900   theme(axis.text.y=element_text(size=16), panel.grid.major.y=element_
        blank(), panel.grid.major.x = element_line())
901
902 speichern("Shared_Mutation_DNA")
903
904 ### RNA ###
905 SM_RNA <- variants_unique %>%
906   filter(SOURCE=="StrelkaRNA") %>%
907   group_by(Patient_ID, Mutation_ID) %>%
908   summarise(N.tumor.per.mut.per.patient=n(), GENE=paste0(unique(GENE),
        collapse = ";"), Metastasis=paste0(unique(Metastasis), collapse="
        ,"), mutationType=paste0(unique(mutationType), collapse = ";"),
```

```
      Effect_group=paste0(unique(Effect_group), collapse = ";")) %>%
909  ungroup() %>% group_by(Mutation_ID) %>%
910  summarise(N.patient.per.mut=n(), N.total.per.mut=sum(N.tumor.per.mut.
      per.patient), GENE=paste0(unique(as.vector(str_split(paste0(GENE,
       collapse = ";"),";", simplify = T))),collapse=";"), mutationType
      =paste0(unique(mutationType), collapse = ";"), Effect_group=
      paste0(unique(Effect_group), collapse = ";"))
911
912 ggplot(SM_RNA %>% slice_max(N.patient.per.mut, n=15), aes(width=.75))+
913  geom_col(aes(x=reorder(Mutation_ID, desc(-N.patient.per.mut)), y=N.
      patient.per.mut), fill=alpha("#bae4b3", 0.4), color="black")+
914  geom_label_repel(size=8, aes(x=reorder(Mutation_ID, desc(-N.patient.
      per.mut)), y=4, hjust = 0, label=GENE))+
915  scale_y_continuous(breaks = scales::breaks_extended(n = 8))+
916  coord_flip()+
917  labs(y="Patients carrying the mutation (count)", x="Mutation ID (with
       corresponding Gene)")+
918  theme_basic()+
919  theme(axis.text.y=element_text(size=16), panel.grid.major.y=element_
      blank(), panel.grid.major.x = element_line())
920
921 speichern("Shared_Mutation_RNA")
922
923
924 SM_DNA_gr <- SM_DNA %>%
925  mutate(temp=1) %>%
926  pivot_wider(names_from = mutationType, values_from = temp) %>%
927  mutate(temp=1) %>%
928  pivot_wider(names_from = Effect_group, values_from = temp) %>% # 4
      Mutationen die rausfallen, vertretbar
929  group_by(N.patient.per.mut) %>%
930  summarise(N.mutations.per.group=n(), GENE=paste0(unique(as.vector(str
      _split(paste0(GENE, collapse = ";"),";", simplify = T))),collapse
      =";"), across(c(4:13), ~ sum(.x, na.rm = T)))
931
932 ggplot(SM_DNA_gr, aes(x=as.factor(N.patient.per.mut), y=N.mutations.per
     .group))+
933  #geom_col(aes(width=.75), fill=alpha("#6495ED", 0.75), color="black")
      +
934  geom_col(aes(width=.75), fill=alpha("#2171b5", 0.55), color="black")+
935  scale_y_continuous(limits=c(0,8000), breaks = c(0, 10, 20, 30, 40,
      160, 180, 200, 7000, 8000))+
936  scale_y_break(c(200,7000), scales = 0.15, space = 0.5)+
937  scale_y_break(c(40,160), scales = 0.25, space = 0.5)+
938  theme_basic()+
939  labs(x="# patients carrying the variants", y="Shared variants (count)
```

```
      ")
940
941 speichern("Shared_Mutation_sets_DNA_2")
942
943 SM_DNA_gr_pie <- SM_DNA_gr %>%
944  mutate(perc = 100*round(N.mutations.per.group/colSums(across(starts_
         with("N.mutations"))),4)) %>%
945  mutate(N.patient.per.mut = ifelse(perc<0.2, "5-14", N.patient.per.mut
         )) %>%
946  group_by(N.patient.per.mut) %>%
947  summarise(perc=sum(perc), N.mutations.per.group=sum(N.mutations.per.
         group)) %>%
948  mutate(csum = rev(cumsum(rev(perc))),
949       pos = perc/2 + lead(csum, 1),
950       pos = if_else(is.na(pos), perc/2, pos),
951       perc_annotaed = round(perc, digits = 1))
952 SM_DNA_gr_pie$annotation <- c(paste0(sum(SM_DNA_gr$N.mutations.per.
        group)), NA, NA, NA, NA)
953
954 cp <- coord_polar(theta = "y")
955 cp$is_free <- function() TRUE
956 ggplot(SM_DNA_gr_pie, aes(x = 1, y = perc, fill = fct_inorder(N.patient
        .per.mut))) +
957  geom_col(width = 0.5, color = 1, alpha=0.8) +
958  geom_label_repel(data = SM_DNA_gr_pie,
959                aes(y = pos, label = paste0(N.mutations.per.group,"\n",
                        perc_annotaed, "%")),
960                size = 12, nudge_x = 1, show.legend = FALSE,
961                box.padding = 0.5,
962                alpha=0.8) +
963  geom_text(aes(x=0.2, y=0, label=annotation), size=14)+
964  cp+
965  scale_fill_brewer(palette = "Blues") +
966  theme_void()+
967  theme_donut()+
968  guides(fill = guide_legend(title = "Patients sharing \nthe variants")
         ) +
969  xlim(c(0.2, 1.5))
970
971 speichern("Shared_Mutation_sets_DNA_Pie_2")
972
973
974 ### RNA ###
975
976 SM_RNA_gr <- SM_RNA %>%
977  mutate(temp=1) %>%
```

```
978   pivot_wider(names_from = mutationType, values_from = temp) %>%
979   mutate(temp=1) %>%
980   pivot_wider(names_from = Effect_group, values_from = temp) %>% # 4
          Mutationen die rausfallen, vertretbar
981   group_by(N.patient.per.mut) %>%
982   summarise(N.mutations.per.group=n(), GENE=paste0(unique(as.vector(str
          _split(paste0(GENE, collapse = ";"),";", simplify = T))),collapse
          =";"), across(c(4:13), ~ sum(.x, na.rm = T)))
983
984 ggplot(SM_RNA_gr, aes(x=as.factor(N.patient.per.mut), y=N.mutations.per
      .group))+
985   geom_col(aes(width=.75), fill=alpha("#bae4b3", 0.75), color="black")+
986   scale_y_break(c(5000,65000), scales = 0.15, space = 0.5)+
987   #scale_y_break(c(1700,3900), scales = 0.25)+
988   scale_y_break(c(1000,1600), scales = 0.25, space = 0.5)+
989   scale_y_continuous(limits=c(0,75000), breaks = c(0,250, 500, 750,
          1000, 2000, 4000, 70000))+
990   theme_basic()+
991   labs(x="# patients sharing the variants", y="Shared variants (count)"
          )
992
993 speichern("Shared_Mutation_sets_RNA")
994
995
996 SM_RNA_gr_pie <- SM_RNA_gr %>%
997   mutate(perc = 100*round(N.mutations.per.group/colSums(across(starts_
          with("N.mutations"))),4)) %>%
998   mutate(N.patient.per.mut = ifelse(perc<1, "5-26", N.patient.per.mut))
          %>%
999   group_by(N.patient.per.mut) %>%
1000  summarise(perc=sum(perc), N.mutations.per.group=sum(N.mutations.per.
          group)) %>%
1001  mutate(csum = rev(cumsum(rev(perc))),
1002         pos = perc/2 + lead(csum, 1),
1003         pos = if_else(is.na(pos), perc/2, pos),
1004         perc_annotaed = round(perc, digits = 1))
1005 SM_RNA_gr_pie$annotation <- c(paste0(sum(SM_RNA_gr$N.mutations.per.
      group)), NA, NA, NA, NA)
1006
1007 cp <- coord_polar(theta = "y")
1008 cp$is_free <- function() TRUE
1009 ggplot(SM_RNA_gr_pie, aes(x = 1, y = perc, fill = fct_inorder(N.patient
      .per.mut))) +
1010   geom_col(width = 0.5, color = 1, alpha=0.8) +
1011   geom_label_repel(data = SM_RNA_gr_pie,
1012                    aes(y = pos, label = paste0(N.mutations.per.group,"\n",
```

```
                    perc_annotaed, "%")),
1013              size = 12, nudge_x = 1, show.legend = FALSE,
1014              box.padding = 0.5,
1015              alpha=0.8) +
1016  geom_text(aes(x=0.2, y=0, label=annotation), size=14)+
1017  cp+
1018  scale_fill_brewer(palette = "Greens") +
1019  theme_void()+
1020  theme_donut()+
1021  guides(fill = guide_legend(title = "Patients sharing \nthe variants")
      ) +
1022  xlim(c(0.2, 1.5))
1023
1024 speichern("Shared_Mutation_sets_RNA_Pie_2")
1025
1026
1027 SM_DNA_gr <- SM_DNA_gr %>%
1028  pivot_longer(cols=c(8:13), names_to = "Effect_group", values_to = "
      value")
1029
1030 # Experimentel
1031
1032 Entity_shared <- variants_unique %>%
1033  filter(SOURCE=="Mutect2") %>%
1034  group_by(Tumor_entity_short, Effect_group) %>%
1035  summarise(count=n()) %>%
1036  ungroup() %>% group_by(Tumor_entity_short) %>%
1037  mutate(test = count/colSums(across(starts_with("count"))))
1038
1039 ################### Mutational Overlap (VENN) with EULERR
      ##############################################################################
1040
1041 for (i in unique((filter(variants_unique, grepl("T2", Master_ID)|grepl(
      "T4", Master_ID)))$Patient_ID)){
1042     euler <- variants_unique %>%
1043       filter(Patient_ID==i) %>%
1044       group_by(Mutation_ID, Metastasis) %>%
1045       summarise(level=paste0(SOURCE, collapse = ";")) %>%
1046       filter(str_detect(level, ";")) %>%
1047       #filter(SOURCE=="Mutect2") %>%
1048       #filter(SOURCE=="StrelkaRNA") %>%
1049       distinct(Mutation_ID, Metastasis, .keep_all = TRUE) %>%
1050       group_by(Metastasis) %>%
1051       summarise(mutation=paste0(Mutation_ID, collapse = ";"), N.
              mutations=n())
```

```
1052      Tu1 <- as.vector(str_split(euler$mutation[1],";", simplify = TRUE)
              )
1053      Tu2 <- as.vector(str_split(euler$mutation[2],";", simplify = TRUE)
              )
1054      Tu4 <- as.vector(str_split(euler$mutation[3],";", simplify = TRUE)
              )
1055
1056      metas <- list(T1=c(Tu1), T2=c(Tu2), T4=c(Tu4))
1057      if(unique(is.na(Tu2))){metas[[2]]=NULL}
1058      if(unique(is.na(Tu4))){metas[[3]]=NULL}
1059
1060      euler(metas)$stress
1061      euler(metas)$diagError
1062
1063      print(plot(euler(metas, shape = "ellipse"),
1064          #counts = list(cex=3, font=7),
1065          quantities = list(type = c("counts", "percent"), font=1, round
                  =1, cex=1.2),
1066          key = TRUE,
1067          #quantities = TRUE,
1068          #percentages = TRUE,
1069          #counts = FALSE,
1070          #fills =list(fill=c(viridis::viridis(n = 3))),
1071          alpha = 0.5,
1072          if(i=="IN_19"){fill = c("lightblue2", "lightsalmon","#99CC99",
                   "plum", "#80B6AB", "#E0BB99", "#A18F8F")}
1073          else{fills = list(fill = c("lightblue2", "lightsalmon","plum2"
                  , "", "", "", ""))},
1074          edges=list(lty = 1),
1075          #factor_names = T,
1076          labels=list(font=2, cex=1.6),
1077          legend = F,
1078          newpage = TRUE
1079          ))
1080
1081      dev.copy(pdf, paste0("plots/Thesis/Variants/euler_DNARNA_", i, ".
              pdf"))
1082
1083      dev.off()
1084  }
1085
1086  #
      ##############################################################################
```

Listing 3: Pipeline for characterization of the mutanome

2.2 Assessment of selection criteria for neoantigen candidates

```
library(tidyverse)
library(openxlsx)
library(stringr)
library(reshape2)
library(ggplot2)
library(extrafont)
library(ggrepel)
library(scales)
library("ggpubr")

source(file = "Peptides/ImmuNeo_peptides_all_V7.R")
source(file = "functions/import.references.R")


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=0.6),
    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.line = element_blank(),
    axis.ticks = element_line(color = "grey"),
    axis.text = element_text(size = 30),
    axis.text.x = element_text(angle = 0),
    axis.title = element_text(size = 30),
    legend.text = element_text(size = 26),
    legend.key.size = unit(1.0, 'cm'),
    legend.title = element_text(size= 30)
  )
}

theme_basic <- function(){
  theme_minimal()+
    theme(axis.title = element_text(size=30),
          axis.text = element_text(size=30),
          axis.text.y.right = element_blank(),
          axis.ticks.x=element_blank(),
          panel.grid.major.x=element_blank(),
          panel.grid.minor=element_blank(),
          legend.key.size = unit(2.5, 'cm'),
```

```
        legend.title = element_text(size=30),
        legend.text = element_text(size=30),
        strip.text.y = element_text(size = 30))
}

extract_duplicates <- function(x){
  duplicates <- list()
  l <- 1
  for (i in (1:length(x)-1)) {
    for (k in ((i+1):length(x))) {
      if (x[k]%in%x[i]){
        duplicates[[l]] <- x[i]
        l <- l+1
      }
    }
  }
  return(duplicates)
}

compare_predictions <- function(x){
  #duplicates <- extract_duplicates(str_sub(colnames(x), 1, 17))
  all.HLA <- as.list(unique(str_sub(colnames(select(x, starts_with("HLA
      "))), 1, 17)))
  for (i in (1:length(all.HLA))) {
    colpair <- str_subset(colnames(x), all.HLA[[i]])
    #varname_Delta <- ifelse(str_detect(colpair[[1]],"rank"), paste0("
        Delta.", str_sub(colpair[[1]],1,17)), paste0("Delta.", str_sub(
        colpair[[1]],1,23)))
    varname_Diff_rel <- ifelse(str_detect(colpair[[1]],"rank"), paste0(
        str_sub(colpair[[1]],1,17),".diff_rel"), paste0(str_sub(colpair
        [[1]],1,23),".diff_rel"))
    #x <- mutate(x,!!varname_Delta := abs(x[[(colpair[[1]])]]-x[[(
        colpair[[2]])]]))
    if(length(colpair)>1){x <- mutate(x,!!varname_Diff_rel := (x[[(
        colpair[[1]])]]-x[[(colpair[[2]])]]) / x[[(colpair[[1]])]])}
    if(str_detect(colpair[[1]],"rank")){
      varname_Binder_type <- paste0("Binder_type.", str_sub(colpair
          [[1]],1,17))
      if(length(colpair)>1){
        x <- mutate(x,!!varname_Binder_type := ifelse(x[[(colpair[[1]])
            ]]<2 |x[[(colpair[[2]])]]<2, ifelse(x[[(colpair[[1]])]]<0.5 |
            x[[(colpair[[2]])]]<0.5, "SB", "WB"), NA))
      }
      else{
        x <- mutate(x,!!varname_Binder_type := ifelse(x[[(colpair[[1]])
            ]]<2 |x[[(colpair[[1]])]]<2, ifelse(x[[(colpair[[1]])]]<0.5 |
```

```
          x[[(colpair[[1]])]]<0.5, "SB", "WB"), NA))
    }
   }
  }
  return(x)
}

predictions_to_long <- function(x){
  x <- x %>%
   select(-(contains("Delta"))) %>%
   pivot_longer(starts_with("HLA")) %>%
   separate(name, into = c("allele", "temp", "BA_type", "predictor"),
       sep = "([\\.])", convert = TRUE) %>%
   select(-temp) %>%
   pivot_longer(contains("Binder"), names_to = "name", values_to = "
       Binder_type") %>%
   separate(name, into = c(NA, "allele_2", NA, NA), sep = "([\\.])",
       convert = TRUE)
  y <- x %>%
   merge(x=select(., (Patient_ID:value)), y=select(., Seq, (allele_2:
       Binder_type)), by.x=c("allele","Seq"), by.y=c("allele_2","Seq"))
        %>%
   distinct(.keep_all = T) %>%
   select(Patient_ID, Master_ID, Seq, allele, everything()) %>%
   mutate_if(is.numeric, round, digits=3)
  return(y)
}

######################

### DELETE MEL15 DATA ###

predictions.all[[25]]=NULL

predictions.all.diff <- lapply(predictions.all, compare_predictions)
predictions.all.diff.long <- lapply(predictions.all.diff, predictions_
    to_long)

# Add "real individual" Binder_type and correct the old binder_type to
    best binder_type
NEW <- bind_rows(predictions.all.diff.long) %>%
  rename("Binder_type_max"=Binder_type) %>%
  mutate(Binder_type=ifelse(BA_type=="rank", ifelse(value<2, ifelse(
      value<0.5, "SB", "WB"),NA), ifelse(BA_type=="prediction",ifelse(
      value<500, ifelse(value<50, "SB", "WB"),NA), NA))) %>%
  mutate(Binder_type=ifelse(predictor=="diff_rel", Binder_type_max,
```

```
      Binder_type))

######################

threshold.method.comparison <- NEW %>%
  filter(predictor=="mhcflurry") %>%
  select(-Binder_type) %>%
  pivot_wider(names_from = BA_type, values_from = value) %>%
  mutate(Binder_type_prediction=ifelse(prediction<500, ifelse(
      prediction<50, "SB", "WB"),NA)) %>%
  mutate(Binder_type_rank=ifelse(rank<2, ifelse(rank<0.5, "SB", "WB"),
      NA)) %>%
  mutate(Binder_type=ifelse(Binder_type_prediction==Binder_type_rank,
      Binder_type_rank, "SB")) %>%
  mutate(Binder_type=ifelse(!is.na(Binder_type_prediction), Binder_type
      _prediction, Binder_type_rank)) %>%
  group_by(allele) %>%
  mutate(N.bindertypes=n()) %>%
  filter(N.bindertypes>16)
threshold.method.comparison$Binder_type[is.na(threshold.method.
    comparison$Binder_type)] <- "NB"

ggplot(data = threshold.method.comparison, aes(x=rank, y=prediction,
    color=allele))+
  geom_point(size=4, aes(shape=Binder_type), alpha=0.7)+
  geom_point(size=2, colour = "white", aes(shape=Binder_type))+
  geom_smooth(span = 0.5, linetype = "dashed", size=0.5, se = F, alpha
      =0.9)+
  theme_PS()+
  theme(legend.position = "right")+
  scale_x_log10(limits=c(min(affinity.comparison.prediction$netMHC),NA)
      , breaks = scales::trans_breaks("log10", function(x) 10^x),
      labels = scales::trans_format("log10", scales::math_format(10^.x)
      ))+
  scale_y_log10()+
  geom_vline(xintercept = 2, color="#02d9d9", linetype="dotdash", size
      =0.7)+
  geom_vline(xintercept = 0.5, color="red", linetype="dotdash", size
      =0.7)+
  geom_hline(yintercept = 500, color="#02d9d9",linetype="dotdash", size
      =0.7)+
  geom_hline(yintercept = 50, color="red",linetype="dotdash", size=0.7)
      +
  labs(x="Percentile Rank", y="Binding affinity [nM]", shape="Binding
      type", color="HLA-allele")

```

```
145 speichern("Test")
146
147 ###################### Good binders vs. best binders (total cohort)
        ##############
148
149 # Number of "good binders" (SB or WB) per allele and per method
150 good.binder.alleles <- NEW %>%
151   mutate(pep.length=nchar(Seq)) %>%
152   filter(predictor=="mhcflurry" & BA_type=="prediction" & (Binder_type
        =="SB"|Binder_type=="WB")) %>%
153   #distinct(Seq, allele, .keep_all = T) %>%
154   group_by(allele) %>%
155   summarise(pep.length.mean=mean(pep.length), N.good_binder=n())
156
157 # Number of "best binders" (BA) (for one prediction algorithm)
158 best.binder.alleles <- IN.10 %>%
159   #distinct(Seq, allele.best.BA.MHCflurry, .keep_all = T) %>%
160   #filter(allele.best.BA.MHCflurry==allele.best.rank.MHCflurry) %>%
161   mutate(pep.length=nchar(Seq)) %>%
162   group_by(allele.best.BA.MHCflurry) %>%
163   summarise(pep.length.mean=mean(pep.length), N.best_binder=n()) %>%
164   rename("allele"=allele.best.BA.MHCflurry)
165
166 allele.comparison <- merge(good.binder.alleles, best.binder.alleles, by
        ="allele", suffixes = c(".good",".best"), all = TRUE) %>%
167   merge(reference.allele.frequency, all.x = TRUE) %>%
168   mutate(allele=str_replace_all(allele,"HLA-", "")) %>%
169   mutate(allele=str_replace_all(allele,c("A"="A*", "B"="B*", "C"="C*"))
        )
170 allele.comparison$N.good_binder[is.na(allele.comparison$N.good_binder)]
        <- 0
171 allele.comparison$N.best_binder[is.na(allele.comparison$N.best_binder)]
        <- 0
172 allele.comparison$allele_frequency[is.na(allele.comparison$allele_
        frequency)] <- 0
173
174 ggplot(allele.comparison, mapping = aes(x=N.good_binder, y=N.best_
        binder))+
175   geom_point(aes(size=allele_frequency))+
176   scale_size(range=c(0.5,8))+
177   geom_label_repel(aes(label=allele))+
178   #geom_text(aes(label=allele), vjust = 0, nudge_y = 0.15, check_
        overlap = TRUE)+
179   #scale_x_continuous(breaks = seq(0,14,2))+
180   #scale_y_continuous(breaks = seq(0,14,2))+
181   theme_PS()+
```

```
182   labs(x="# of associated WB & SB (total cohort)", y="# peptides
          yielding best binding allele", size="Allele Frequency")
183
184 speichern("test")
185
186 ###################### # of Peptide candidates per HLA-allele (within
        total cohort) vs. allele frequency #############
187
188 peptide.per.allele <- NEW %>%
189   filter(BA_type=="prediction") %>%
190   filter(predictor=="mhcflurry") %>%
191   group_by(allele) %>%
192   mutate(N.peptide.per.allele=n()) %>%
193   group_by(allele, Binder_type) %>%
194   summarize(N.peptide.per.allele.Binder_type=n(), N.peptide.per.allele=
          max(N.peptide.per.allele)) %>%
195   merge(reference.allele.frequency, all.x = TRUE) %>%
196   mutate_at(c("allele_frequency"), as.numeric) %>%
197   ungroup() %>%
198   mutate(allele=str_replace_all(allele,"HLA-", "")) %>%
199   mutate(allele=str_replace_all(allele,c("A"="A-", "B"="B-", "C"="C-"))
          ) %>%
200   group_by(allele) %>%
201   mutate(N.bindertypes=n()) %>%
202   filter(N.peptide.per.allele>4)
203
204 coeff <- 0.02
205
206 ggplot(data=peptide.per.allele, aes(x=reorder(allele, desc(N.peptide.
        per.allele))))+
207   geom_bar(aes(fill=Binder_type, y=N.peptide.per.allele.Binder_type),
          position="stack", stat="identity")+
208   geom_col(aes(y=(allele_frequency/N.bindertypes)/coeff))+
209   geom_text(aes(label=round(allele_frequency, digits = 2), y = 2),
          color="white", size=5, angle=90)+
210   theme_PS()+
211   theme(axis.text.x = element_text(angle = 45))+
212   scale_y_continuous(
213     # Features of the first axis
214     name = "# of peptide candidates (within total cohort)",
215
216     # Add a second axis and specify its features
217     sec.axis = sec_axis(~.*coeff, name="Allele frequency", breaks = seq
            (0,1,0.1))
218   )+
219   labs(fill="Binding Type", x="MHC allele")
```

```
220
221 ###################### Directly compare ranks MHCflurry vs. netMHC for
      WB and SB considering the allele frequency
    ########################
222
223 affinity.comparison.rank <- NEW %>%
224   filter(BA_type=="rank" & predictor!="diff_rel") %>%
225   #filter(!is.na(Binder_type_max)) %>%
226   #filter(Binder_type_max=="WB") %>%
227   #filter(value<2) %>%
228   select(-Binder_type) %>%
229   pivot_wider(names_from = predictor, values_from = value) %>%
230   merge(reference.allele.frequency, all.x = TRUE)
231   #filter((mhcflurry*netMHC)<5)
232 affinity.comparison.rank$allele_frequency[is.na(affinity.comparison.
      rank$allele_frequency)] <- min(affinity.comparison.rank$allele_
      frequency, na.rm = T)
233
234
235 affinity.comparison.prediction <- NEW %>%
236   filter(BA_type=="prediction" & predictor!="diff_rel") %>%
237   #filter(!is.na(Binder_type_max)) %>%
238   #filter(value<2) %>%
239   select(-Binder_type) %>%
240   pivot_wider(names_from = predictor, values_from = value) %>%
241   merge(reference.allele.frequency, all.x = TRUE)
242 affinity.comparison.prediction$allele_frequency[is.na(affinity.
      comparison.prediction$allele_frequency)] <- min(affinity.comparison
      .prediction$allele_frequency, na.rm = T)
243
244 ### match affinity.comparison with actual peptide list
245
246 NACs <- unique(DF.plot$Seq)
247 affinity.comparison.rank <- affinity.comparison.rank %>%
248   filter(Seq %in% NACs)
249 affinity.comparison.prediction <- affinity.comparison.prediction %>%
250   filter(Seq %in% NACs)
251
252 ggplot(affinity.comparison.rank, mapping = aes(x=mhcflurry, y=netMHC))+
253   geom_point(aes(size=allele_frequency))+
254   scale_size(range=c(0.1,6))+
255   geom_abline(intercept = 0, slope = 1, color="red", linetype="dashed")
        +
256   geom_smooth(method = "lm", se=TRUE, span=100)+
257   scale_x_log10(limits=c(0.0005,NA), breaks = scales::trans_breaks("
        log10", function(x) 10^x), labels = scales::trans_format("log10",
```

```
      scales::math_format(10^.x)))+
  scale_y_log10(limits=c(0.0005,NA), breaks = scales::trans_breaks("log10", function(x) 10^x), labels = scales::trans_format("log10", scales::math_format(10^.x)))+
  theme_PS()+
  labs(x="Prediction rank MHCflurry", y="Prediction rank netMHC", size="Allele Frequency")

speichern("Rank_comparison_all_pairings_1")

ggplot(affinity.comparison.prediction, mapping = aes(x=mhcflurry, y=netMHC))+
  geom_point(aes(size=allele_frequency))+
  scale_size(range=c(0.1,6))+
  geom_abline(intercept = 0, slope = 1, color="red")+
  geom_smooth(method = "lm", se=F)+
  scale_x_log10(limits=c(min(affinity.comparison.prediction$netMHC),NA), breaks = scales::trans_breaks("log10", function(x) 10^x), labels = scales::trans_format("log10", scales::math_format(10^.x)))+
  scale_y_log10(limits=c(min(affinity.comparison.prediction$netMHC),NA), breaks = scales::trans_breaks("log10", function(x) 10^x), labels = scales::trans_format("log10", scales::math_format(10^.x)))+
  theme_PS()+
  labs(x="Binding affinity prediction MHCflurry (Kd [nM])", y="Bindng affinity prediction rank netMHC (Kd [nM])", size="Allele Frequency")

##### Spearman/Pearson - Correlation between netMHC and MHCflurry #############

ggscatter(affinity.comparison.rank, x = "mhcflurry", y = "netMHC",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.coef.size = 10, cor.method = "pearson",
          xlab = "MHCflurry", ylab = "netMHC", show.legend = FALSE, size=5
          #, fill = "Binder_type_max"
          )+
          geom_abline(intercept = 0, slope = 1, color="red", linetype = "dashed")+
          #geom_point(aes(color=Binder_type_max), size=2)+
          geom_point(aes(color=allele_frequency), size=4)+
          scale_color_viridis_c(alpha = 1, begin=0.2, end=1, breaks=c(0,0.05,0.10,0.15,0.20,0.25), limits=c(0,0.2), na.value = "yellow")+
```

```
286         #yscale("log10", .format = TRUE)+
287         #xscale("log10", .format = TRUE)+
288         scale_x_log10(limits=c(min(affinity.comparison.prediction$
                netMHC),NA), breaks = scales::trans_breaks("log10",
                function(x) 10^x), labels = scales::trans_format("log10",
                scales::math_format(10^.x)))+
289         scale_y_log10(limits=c(0.01,NA), breaks = scales::trans_breaks(
                "log10", function(x) 10^x), labels = scales::trans_format("
                log10", scales::math_format(10^.x)))+
290         theme_PS()+
291         #scale_color_hue(labels = c("SB", "WB", "No binder"))+
292         theme(legend.key.size = unit(2.0, 'cm'), legend.position = "
                right")+
293         guides(fill="none")+
294         #guides(colour = guide_legend(override.aes = list(size=10)))+
295         labs(x="Percentile rank MHCflurry", y="Percentile rank netMHC",
296             #color="Binding type",
297             color="Allele frequency\n(reference pop.)",
298             fill=""
299             )
300
301 speichern("pearson_correlation_rank_all_pairings_mod_16")
302
303 ggscatter(affinity.comparison.prediction, x = "mhcflurry", y = "netMHC"
        ,
304         add = "reg.line", conf.int = TRUE,
305         cor.coef = TRUE, cor.coef.size = 10, cor.coef.coord = c(3,1.5),
                cor.method = "pearson",
306         xlab = "MHCflurry", ylab = "netMHC", size=3)+
307         #geom_abline(intercept = 0, slope = 1, color="red", linetype =
                "dashed")+
308         geom_point(aes(color=Binder_type_max), size=2)+
309         #yscale("log10", .format = TRUE)+
310         #xscale("log10", .format = TRUE)+
311         scale_x_log10(limits=c(11,NA), breaks = scales::trans_breaks("
                log10", function(x) 10^x), labels = scales::trans_format("
                log10", scales::math_format(10^.x)))+
312         scale_y_log10(limits=c(min(affinity.comparison.prediction$
                netMHC),NA), breaks = scales::trans_breaks("log10",
                function(x) 10^x), labels = scales::trans_format("log10",
                scales::math_format(10^.x)))+
313         theme_PS()+
314         scale_color_hue(labels = c("SB", "WB", "No binder"))+
315         theme(legend.key.size = unit(2.0, 'cm'), legend.position = "
                right")+
316         guides(colour = guide_legend(override.aes = list(size=10)))+
```

```
        geom_segment(x=2.699, y=Inf, xend=2.699, yend=-Inf, colour="#02
            d9d9", alpha=1, linetype="dotdash", size=0.4)+
        geom_segment(x=Inf, y=2.699, xend=-Inf, yend=2.699, colour="#02
            d9d9", alpha=1, linetype="dotdash", size=0.4)+
        geom_segment(x=1.699, y=Inf, xend=1.699, yend=-Inf, colour="red
            ", alpha=1, linetype="dotdash", size=0.4)+
        geom_segment(x=Inf, y=1.699, xend=-Inf, yend=1.699, colour="red
            ", alpha=1, linetype="dotdash", size=0.4)+
        #geom_rect(xmin=0, ymin=0, xmax=2.6999, ymax=2.6999, alpha=0.2,
             fill="red")+
        labs(x="Binding affinity MHCflurry (Kd [nM])", y="Bindng
            affinity netMHC (Kd [nM])", color="Binding Type\n(rank
            method)")

speichern("pearson_correlation_affinity_all_pairings_mod_5")

#######################

# Save Plot
save.path = "Peptides/plots/20_08_Prediction_Analysis/"
speichern <- function(name){
  save.filename = paste(save.path,name, ".pdf", sep = "")
  ggsave(save.filename, width = 20, height = 10, dpi = "retina")
}
```

Listing 4: Pipeline for the assessment of selection criteria for neoantigen candidates

2.3 Specifications of neoantigen candidates

```
# import Tidyverse
library(tidyverse)
library(openxlsx)
library(stringr)
library(reshape2)
library(ggplot2)
library(extrafont)


################## DOCUMENTATION ###################
####################################################
# Different DF evolve from selection process

### IN.all.peptides: all peptides imported from the 2 .tsv-files for
```

```
    PROSIT and pFind respectively
   # (*) DISCARD comment, REF, ALT, multiEntity, multiPatient,
       multiGene
   # (*) SUM collapsed values for the 8 cols: TumorAD.Mutect2 etc.

### IN.1.all :
   # (*) Patient_ID reference, (*) Tumor_entity + Metastatic_site

### IN.2.base:
   # (*) TumorVF values

### IN.2.avail:
   # (*) All peptides from patients with HLA-Alleles available in MHC_
       flurry

### IN.3 + IN.4:
   # (*) DFs with sequences for processing in Python with MHC_flurry

### IN.6.list & IN.6.list.unique:
   # (*) DETAILED List with Predictions for all alleles
   # (*) BA.best for best allele
   # (*) unique: eliminated all doubles within one patient

### IN.7.list & IN.7.list.unique:
   # (*) SUMMARY List with BA.best and most important parameters for
       evaluation


### IN.9.Order:
   # (*) INCLUDES BLAST-Information (n_identicals, hit_loci, e_value)
   # (*) EXCLUDES all peptides with more than 2 blast hits

### IN.7.NEW.HLA.C.WB.BA / IN.7.NEW.HLA.C.WB.rank:
   # (*) Peptides that have significantly improved BA / rank through
       consideration of HLA-C


# import all .tsv files from folder and add to one dataframe
# path.tsv.files="Peptides/rawfiles/tsv/"
# tsv.files=list.files(path=path.tsv.files, pattern = "pFind_
    allPatients_hitsAll_processed", full.names = T)
# IN.all.peptides.pFind <- plyr::ldply(tsv.files, read.delim, na.
    strings = c("","NA")) %>%
# filter(TrueHit=="maybe" | TrueHit=="yes")

## Load combined file (.tsv):
```

```
IN.all.peptides.combined <- read_csv("Peptides/rawfiles/tsv/allPatients_hitsAll_Pfind-PROSIT_forPhilipp_2.tsv") %>%
# IN.all.peptides.combined <- read_csv("Peptides/rawfiles/tsv/allPatients_hitsAll_MaxQ-PROSIT_forPhilipp_old.tsv") %>%
  filter(TrueHit=="maybe" | TrueHit=="yes")


## Load pFind file (.tsv):
IN.all.peptides.pFind <- read_tsv("Peptides/rawfiles/tsv/allPatients_hitsAll_processed_pFind.tsv") %>%
  filter(TrueHit=="maybe" | TrueHit=="yes")
IN.all.peptides.pFind["MS.tool"] <- "pFind"
#IN.all.peptides.pFind.2["batch.pipeline"] <- "2"


## Load PROSIT file (.tsv) -- only the one with peptideScore --:
IN.all.peptides.Prosit <- read_tsv("Peptides/rawfiles/tsv/allPatients_FDR005_hitsAll_processed_PROSIT_peptideScore.tsv") %>%
  filter(TrueHit=="maybe" | TrueHit=="yes") %>%
  mutate(patientID=str_sub(patientID, 1, 9))
IN.all.peptides.Prosit["MS.tool"] <- "PROSIT"
#IN.all.peptides.Prosit["batch.pipeline"] <- "1"


# function to bind two dfs with different cols (and different order of cols)
rbind.match.columns <- function(input1, input2) {
  n.input1 <- ncol(input1)
  n.input2 <- ncol(input2)
  names.union <- union(names(input1),names(input2))
  input1[c(setdiff(names.union,names(input1)))] <- NA
  input2[c(setdiff(names.union,names(input2)))] <- NA

  if (n.input2 < n.input1) {
    TF.names <- which(names(input2) %in% names(input1))
    column.names <- names(input2[, TF.names])
  } else {
    TF.names <- which(names(input1) %in% names(input2))
    column.names <- names(input1[, TF.names])
  }

  return(rbind(input1[, column.names], input2[, column.names]))
}

IN.all.peptides <- rbind.match.columns(IN.all.peptides.Prosit, IN.all.peptides.pFind) %>%
```

```
94   select(-comment, -multiEntity, -multiPatient, -multiGene) %>%
95   select(patientID, CHROM, POS, Seq, SeqMarked, SeqGroup, gene, MS.tool
         , calledBy, TrueHit, nReps, nFound, mutationType, transcriptTypes
         , geneBiotype, transcriptBiotype, EFFECT, scoreMS, everything())
96
97 # use new combined file (Niklas) for further analysis
98 IN.all.peptides <- IN.all.peptides.combined %>%
99   rename(MS.tool=quantBy)
100
101
102 # function to sum all collapsed semicolon-separated values in a certain
       column
103 sum.collapsed <- function(x){
104   sum(as.numeric(unlist(strsplit(as.character(x),";",fixed = T))))
105 }
106
107 # old
108 # IN.all.peptides<- data.frame(IN.all.peptides[1:19], apply(IN.all.
      peptides[20:27],c(1,2),sum.collapsed),IN.all.peptides[28:ncol(IN.
      all.peptides)])
109
110 # adapted to niklas' new data
111 IN.all.peptides<- data.frame(IN.all.peptides[1:18], apply(IN.all.
      peptides[19:26],c(1,2),sum.collapsed),IN.all.peptides[27:ncol(IN.
      all.peptides)])
112
113
114 # IMPORT REFERENCES
115 source(file = "functions/import.references.R")
116
117 # change name of first column and add col "Master_ID_group" by use of
       str_sub
118 colnames(IN.all.peptides)[1] <- "Master_ID"
119 IN.all.peptides$Master_ID_group <- as.factor(str_sub(IN.all.peptides$
      Master_ID,1,6))
120 IN.1.all <- select(IN.all.peptides, Master_ID, Master_ID_group,
      everything())
121 # Renaming rows etc.
122 IN.1.all <- merge(reference.master, IN.1.all, by.x = "Master_ID", by.y
      = "Master_ID_group") %>%
123   select(-Master_ID) %>%
124   rename(Master_ID = Master_ID.y) %>%
125   merge(reference.entity) %>%
126   select(Patient_ID, Master_ID, everything())
127
128 #### IN.2: Selection of rows ####
```

```
IN.2.base <- IN.1.all %>%
  mutate(total_reads.StrelkaRNA = TumorAD.StrelkaRNA+TumorRD.StrelkaRNA) %>%
  mutate(total_reads.Mutect2 = TumorAD.Mutect2+TumorRD.Mutect2) %>%
  mutate(TumorVF.StrelkaRNA = TumorAD.StrelkaRNA / (TumorAD.StrelkaRNA+TumorRD.StrelkaRNA)) %>%
  mutate(TumorVF.Mutect2 = TumorAD.Mutect2 / (TumorAD.Mutect2+TumorRD.Mutect2)) %>%
  select(Patient_ID,Master_ID,CHROM,POS,Seq,gene,Tumor_entity,calledBy,nReps,nFound,transcriptTypes,TumorVF.Mutect2,TumorVF.StrelkaRNA,total_reads.Mutect2,total_reads.StrelkaRNA,TumorAD.Mutect2,TumorRD.Mutect2,TumorAD.StrelkaRNA,TumorRD.StrelkaRNA,everything()) %>%
  arrange(Patient_ID)
IN.2.base.export <- IN.2.base %>%
  select(Patient_ID, Master_ID, Seq) %>%
  distinct(Patient_ID, Master_ID, Seq)

# available.alleles-reference
alleles_available = unlist(c(reference.alleles.available))
alleles_available.netMHC = unlist(c(reference.alleles.available.netMHC))

process_df_for_netMHC <- function(input_DF, Seq_column, reference.HLA){
  output <- input_DF %>%
    merge(reference.HLA) %>%
    pivot_longer(-(Patient_ID:Seq_column), names_to = "allele_nr", values_to = "allele", values_drop_na = T) %>%
    rename(peptide=Seq_column) %>%
    arrange(Patient_ID) %>%
    mutate(netMHC.available = str_detect(allele, paste0(alleles_available.netMHC, collapse = "|")), id=Patient_ID, allele=as.factor(allele)) %>%
    select(id, Patient_ID, everything()) %>%
    filter(netMHC.available==T) %>%
    select(-id, -netMHC.available)
}

process_df_for_prediction_all <- function(input_DF, Seq_column, reference.HLA){
  output <- input_DF %>%
    merge(reference.HLA) %>%
    pivot_longer(-(Patient_ID:Seq_column), names_to = "allele_nr", values_to = "allele", values_drop_na = T) %>%
    rename(peptide=Seq_column) %>%
    arrange(Patient_ID)
```

```
}

# IN.2.exported.pred_avail <- process_df_for_prediction_available(IN.2.base.export, Seq_column = "Seq", reference.HLA = reference.HLA)
IN.2.exported.netMHC <- process_df_for_netMHC(IN.2.base.export, Seq_column = "Seq", reference.HLA = reference.HLA)
IN.2.exported.all <- process_df_for_prediction_all(IN.2.base.export, Seq_column = "Seq", reference.HLA = reference.HLA)

# Export .csv file for MHCflurry

for (i in unique(IN.2.exported.all$Patient_ID)){
  IN.2.exported.all.selected <- IN.2.exported.all %>%
    filter(Patient_ID==i)
  write.csv(IN.2.exported.all.selected, file = paste0("../../Python/input_files/mut_peptides/all/", i, ".csv"), row.names=FALSE)
}


# Export Fastas for netMHC
IN.2.exported.netMHC.peptides <- IN.2.exported.netMHC %>%
  distinct(Patient_ID, peptide, .keep_all = F) %>%
  group_by(Patient_ID) %>%
  #mutate(Seq_ID=letters[row_number()]) %>%
  mutate(Seq_ID=sprintf("%03d", row_number())) %>%
  mutate(Seq_ID=paste0(Patient_ID,sep="_",Seq_ID,sep="_",peptide)) %>%
  select(Seq_ID,everything()) %>%
  mutate(Seq_ID=paste0(sep=">",Seq_ID)) %>%
  mutate(Seq=as.character(peptide)) %>%
  ungroup() %>%
  select(Patient_ID,Seq_ID,peptide) %>%
  as.data.frame()

for (i in unique(IN.2.exported.netMHC.peptides$Patient_ID)){
  IN.2.exported.netMHC.peptides.single <- IN.2.exported.netMHC.peptides %>%
    filter(Patient_ID==i) %>%
    select(Seq_ID,peptide)
  temp <- data.frame(x=1:(2*nrow(IN.2.exported.netMHC.peptides.single)))
  for (x in temp$x) {
    temp[x,2] <- c(IN.2.exported.netMHC.peptides.single[floor((x-1)/2)+1,((x+1)%%2)+1])
  }
  IN.2.exported.netMHC.peptides.single <- temp %>%
    select(-x)
```

```
  write_csv(IN.2.exported.netMHC.peptides.single, path = paste0("../../Python/input_files/mut_peptides/preselected_for_netMHC/", i, ".fasta"), col_names = F)
}

# Export HLA alleles for netMHC
IN.2.exported.netMHC.alleles <- IN.2.exported.netMHC %>%
  distinct(Patient_ID, allele, .keep_all = T) %>%
  select(Patient_ID, allele_nr, allele) %>%
  pivot_wider(names_from = allele_nr, values_from = allele) %>%
  select(Patient_ID, "HLA-A.1", "HLA-A.2", "HLA-B.1", "HLA-B.2", "HLA-C.1", "HLA-C.2")

write_csv(IN.2.exported.netMHC.alleles, path = "../../Python/input_files/mut_peptides/preselected_for_netMHC/alleles.csv", col_names = F)

############ DO PREDICTIONS WITH MHC_FLURRY / NETMHC ##############

#### Import predictions (MHCflurry) as DF ####
path.predictions.mut <- dir('../../Python/result_files/mut_peptides/models_class1_presentation/', pattern="*.csv", full.names=TRUE)
#path.predictions.mut <- dir('../../Python/result_files/mut_peptides/models_class1_pan/', pattern="*.csv", full.names=TRUE) #for old results (old peptides)
predictions.mhcflurry.long <- lapply(path.predictions.mut, read.csv)

#### Import predictions (netMHC) as DF ####
path.predictions.mut <- dir('../../Python/result_files/mut_peptides/netMHC/tsv/', pattern="*.tsv", full.names=TRUE)
predictions.netMHC.raw <- lapply(path.predictions.mut, read_tsv)

# reshape predictions for each DF in list (MHCflurry)
process.predictions <- function(x)
  {
  select(x, Patient_ID, Master_ID, allele, peptide, mhcflurry_affinity, mhcflurry_affinity_percentile) %>%
    group_by(Patient_ID, Master_ID, allele, peptide) %>%
    summarise(BA.prediction.mhcflurry=mean(mhcflurry_affinity), BA.percentile.mhcflurry=mean(mhcflurry_affinity_percentile)) %>%
    gather(key=mhcflurry_type, value=prediction.value, -(Patient_ID:peptide)) %>%
    unite(temp, allele, mhcflurry_type, sep = ".") %>%
    spread(temp, prediction.value) %>%
    rename(Seq="peptide")
}
```

```
235
236 process.predictions_old <- function(x)
237 {
238   select(x, Patient_ID, Master_ID, allele, peptide, mhcflurry_
        prediction, mhcflurry_prediction_percentile) %>%
239    group_by(Patient_ID, Master_ID, allele, peptide) %>%
240    summarise(BA.prediction.mhcflurry=mean(mhcflurry_prediction), BA.
        percentile.mhcflurry=mean(mhcflurry_prediction_percentile)) %>%
241    gather(key=mhcflurry_type, value=prediction.value, -(Patient_ID:
        peptide)) %>%
242    unite(temp, allele, mhcflurry_type, sep = ".") %>%
243    spread(temp, prediction.value) %>%
244    rename(Seq="peptide")
245 }
246
247 # reshape predictions for each DF in list (netMHC)
248 process.predictions.netMHC <- function(x){
249   cols_with_BA.pred = grep("HLA",colnames(x))
250   temp <- x %>%
251    rename_at(vars(starts_with("HLA")), funs(str_replace(.,"$",".BA.
        prediction.netMHC"))) %>%
252    rename_at(vars(cols_with_BA.pred+1), funs(colnames(x)[grep("HLA",
        colnames(x))])) %>%
253    rename_at(vars(cols_with_BA.pred+1), funs(str_replace(.,"$",".BA.
        percentile.netMHC"))) %>%
254    slice(2:nrow(.)) %>%
255    rename(Seq = ...2) %>%
256    mutate(Patient_ID=str_sub(...3,1,5)) %>%
257    select(Patient_ID,Seq,everything(),-contains("...")) %>%
258    mutate_at(vars(c(3:ncol(.))), as.numeric)
259   distinct(temp)
260 }
261
262 # DEBUG ####################
263 test <- predictions.netMHC.raw[[1]]
264 cols_with_BA.pred = grep("HLA", colnames(test))
265
266 testneu <- test %>%
267   rename_at(vars(starts_with("HLA")), funs(str_replace(.,"$",".BA.
        prediction.netMHC"))) %>%
268   rename_at(vars(cols_with_BA.pred+1), funs(colnames(test)[grep("HLA",
        colnames(test))])) %>%
269   rename_at(vars(cols_with_BA.pred+1), funs(str_replace(.,"$",".BA.
        percentile.netMHC"))) %>%
270   slice(2:nrow(.)) %>%
271   rename(Seq = ...2) %>%
```

```
272  mutate(Patient_ID=str_sub(...3,1,5)) %>%
273  select(Patient_ID,Seq,everything(),-contains("...")) %>%
274  mutate_at(vars(c(3:ncol(.))), as.numeric)
275
276 reference.alleles.available.test <- read_csv2("rawfiles/references/
        available.alleles_reference_3.csv")
277
278 test <- setdiff(reference.alleles.available, reference.alleles.
        available.test)
279
280 ##########################
281
282 predictions.mhcflurry.wide <- lapply(predictions.mhcflurry.long,
        process.predictions)
283 #predictions.mhcflurry.wide <- lapply(predictions.mhcflurry.long,
        process.predictions_old) #for old results (old peptides)
284 predictions.netMHC.wide <- lapply(predictions.netMHC.raw, process.
        predictions.netMHC)
285
286 # Merge both predictions & sort and select
287 predictions.all.raw <- purrr::map2(predictions.mhcflurry.wide,
        predictions.netMHC.wide, merge)
288
289 sort_and_select.predictions <- function(x){
290  temp <- x %>%
291    rename_at(vars(1:ncol(.)), funs(str_replace(., "percentile", "rank")
           )) %>%
292    select(sort(names(.))) %>%
293    select(Patient_ID, Master_ID, Seq, everything()) %>%
294    mutate_if(is.numeric, round, digits=4)
295 }
296
297 predictions.all <- lapply(predictions.all.raw, sort_and_select.
        predictions)
298
299 # DF with available predictions
300 IN.2.avail <- IN.2.base %>%
301  merge(reference.HLA) %>%
302  unite("allele", -(1:(ncol(.)-6)), sep = ",") %>%
303  mutate(mhcflurry.available = str_detect(allele, paste0(alleles_
        available, collapse = "|")), id=Patient_ID, allele=as.factor(
        allele)) %>%
304  mutate(netMHC.available = str_detect(allele, paste0(alleles_available
        .netMHC, collapse = "|")), id=Patient_ID, allele=as.factor(allele
        )) %>%
305  filter(netMHC.available==T) %>%
```

```
306   select(-id)
307
308
309 #### IN.5: Divide IN.2.selection into elements in a list
310 IN.5.list.all <- split(IN.2.base, f=IN.2.base$Patient_ID)
311 IN.5.list.avail <- split(IN.2.avail, f=IN.2.avail$Patient_ID)
312
313 #### IN.6: Merge predictions with IN.5.list.avail into new List
314 IN.6.list.raw <- purrr::map2(IN.5.list.avail, predictions.all, merge,
      by.x=c("Seq","Master_ID"), by.y=c("Seq","Master_ID"))
315
316 # Modify DFs in IN.6.list: Define functions for lists - final list (6)
      and plot(7)
317
318 find_best_binder <- function(x) {
319   temp <- rename(x, Patient_ID=Patient_ID.x) %>%
320     select(-Patient_ID.y) %>%
321     select(Patient_ID, Master_ID, everything()) %>%
322     mutate(BA.best=do.call(pmin, x[grep("BA.prediction",colnames(x))])
          %>%
323     mutate(BA.rank.best=do.call(pmin, x[grep("BA.percentile",colnames(x)
          )])) %>%
324     mutate(allele.BA.best=colnames(x[grep("BA.prediction",colnames(x))])
          [apply(x[grep("BA.prediction",colnames(x))],1,which.min)]) %>%
325     mutate(allele.BA.best=substr(allele.BA.best, 1,9)) %>%
326     mutate(allele.rank.best=colnames(x[grep("BA.percentile",colnames(x))
          ])[apply(x[grep("BA.percentile",colnames(x))],1,which.min)]) %>%
327     mutate(allele.rank.best=substr(allele.rank.best, 1,9))
328 }
329
330 find_best_binder_2 <- function(x) {
331   temp <- rename(x, Patient_ID=Patient_ID.x) %>%
332     select(-Patient_ID.y) %>%
333     select(Patient_ID, Master_ID, everything()) %>%
334     mutate(BA.best.MHCflurry=do.call(pmin, x[grep("BA.prediction.
          mhcflurry",colnames(x))])) %>%
335     mutate(BA.best.netMHC=do.call(pmin, x[grep("BA.prediction.netMHC",
          colnames(x))])) %>%
336     mutate(Rank.best.MHCflurry=do.call(pmin, x[grep("BA.rank.mhcflurry",
          colnames(x))])) %>%
337     mutate(Rank.best.netMHC=do.call(pmin, x[grep("BA.rank.netMHC",
          colnames(x))])) %>%
338     mutate(allele.best.BA.MHCflurry=colnames(x[grep("BA.prediction.
          mhcflurry",colnames(x))])[apply(x[grep("BA.prediction.mhcflurry"
          ,colnames(x))],1,which.min)]) %>%
339     mutate(allele.best.BA.MHCflurry=substr(allele.best.BA.MHCflurry,
```

```
        1,9)) %>%
340   mutate(allele.best.BA.netMHC=colnames(x[grep("BA.prediction.netMHC",
        colnames(x))])[apply(x[grep("BA.prediction.netMHC",colnames(x))
        ],1,which.min)]) %>%
341   mutate(allele.best.BA.netMHC=substr(allele.best.BA.netMHC, 1,9)) %>%
342   mutate(allele.best.rank.MHCflurry=colnames(x[grep("BA.rank.mhcflurry
        ",colnames(x))])[apply(x[grep("BA.rank.mhcflurry",colnames(x))
        ],1,which.min)]) %>%
343   mutate(allele.best.rank.MHCflurry=substr(allele.best.rank.MHCflurry,
         1,9)) %>%
344   mutate(allele.best.rank.netMHC=colnames(x[grep("BA.rank.netMHC",
        colnames(x))])[apply(x[grep("BA.rank.netMHC",colnames(x))],1,
        which.min)]) %>%
345   mutate(allele.best.rank.netMHC=substr(allele.best.rank.netMHC, 1,9))
         %>%
346   left_join(reference.allele.frequency, by=c("allele.best.BA.MHCflurry
        "="allele")) %>%
347   rename("allele.frequency.MHCflurry"=allele_frequency) %>%
348   left_join(reference.allele.frequency, by=c("allele.best.BA.netMHC"="
        allele")) %>%
349   rename("allele.frequency.netMHC"=allele_frequency)
350 }
351
352 sum_up_DF <- function(x) {
353  temp <- select(x, Patient_ID, Master_ID, Seq, SeqMarked, CHROM, POS,
       gene, TrueHit, calledBy, MS.tool, Tumor_entity, Tumor_entity_
       short, Tumor_state, Metastatic_site, Tumor_origin, total_reads.
       StrelkaRNA, total_reads.Mutect2, TumorVF.StrelkaRNA, TumorVF.
       Mutect2, BA.best.MHCflurry, BA.best.netMHC, allele.best.BA.
       MHCflurry, allele.best.BA.netMHC, Rank.best.MHCflurry, Rank.best.
       netMHC, allele.best.rank.MHCflurry, allele.best.rank.netMHC,
       transcriptTypes, mutationType, transcriptBiotype, geneBiotype,
       EFFECT, scoreMS, scorePeptide, allele.frequency.MHCflurry, allele
       .frequency.netMHC, header)
354 }
355
356 #### Final Lists (with double entries of peptides if i) resulting from
      different genes, ii) resulting from different metastasis)####
357 # IN.6: complete list with all BA-predictions for all alleles
      respectively
358 # IN.7: summary list of basic information with BA.best
359 IN.6.list <- lapply(IN.6.list.raw, find_best_binder_2)
360 IN.7.list <- lapply(IN.6.list, sum_up_DF)
361
362 #### Unique peptide selection (unique in (Patient,Seq) - may contain
      double multilets-Seq from different patients); will "distinct" rows
```

```
    with the same Peptide)
modify_DF_unique <- function(x) {
  temp1 <- x %>%
    mutate(Master_ID_group=as.factor(str_sub(Master_ID,1,6))) %>%
    select(Patient_ID, Master_ID_group, everything()) %>%
    distinct(Patient_ID, Master_ID_group, Seq, .keep_all = T) %>%
    select(-Master_ID,-TumorVF.StrelkaRNA,-TumorVF.Mutect2,-MS.tool, -
        CHROM, -POS, -scoreMS, -scorePeptide)
  temp2 <- x %>%
    mutate(Master_ID_group=as.factor(str_sub(Master_ID,1,6))) %>%
    select(Patient_ID, Master_ID_group, everything()) %>%
    group_by(Patient_ID, Master_ID_group, Seq) %>%
    summarise(Metastasis = paste0(str_sub(Master_ID,8,9),collapse = ";")
        , MS.tool = paste0(sort(MS.tool), collapse = ";") , TumorVF.
        StrelkaRNA = max(TumorVF.StrelkaRNA), TumorVF.Mutect2 = max(
        TumorVF.Mutect2), CHROM = paste0(CHROM, collapse = ";"), POS =
        paste0(POS, collapse = ";"), scoreMS=min(scoreMS, na.rm = T),
        scorePeptide=max(scorePeptide)) %>%
    mutate(scoreMS=ifelse(is.infinite(scoreMS), NA, scoreMS))
    # alternatively: group_by(Patient_ID, Master_ID_group, Seq) %>%
    # summarise_all(list( ~ paste0(.,collapse = " + ")))
  temp2["MS.tool"] <- unlist(lapply(temp2$MS.tool, function(x) {paste0(
      unique(as.vector(str_split(x,";", simplify = T))),collapse=" + ")
      } ))
  temp2["Metastasis"] <- unlist(lapply(temp2$Metastasis, function(x) {
      paste0(unique(as.vector(str_split(x,";", simplify = T))),collapse
      =" + ")} ))
  temp2["Chrom"] <- unlist(lapply(temp2$CHROM, function(x) {paste0(
      unique(as.vector(str_split(x,";", simplify = T))),collapse=" + ")
      } ))
  temp2["Pos"] <- unlist(lapply(temp2$POS, function(x) {paste0(unique(
      as.vector(str_split(x,";", simplify = T))),collapse=" + ")} ))
  temp <- merge(temp1, temp2) %>%
    select(Patient_ID, Master_ID_group, Metastasis, Seq, SeqMarked,
        Chrom, Pos, gene, TrueHit, MS.tool, everything()) %>%
    select(-CHROM,-POS) %>%
    select(-header, everything()) %>%
    mutate_if(is.numeric, round, digits=6)
}

IN.6.list.unique <- lapply(IN.6.list, modify_DF_unique)
IN.7.list.unique <- lapply(IN.7.list, modify_DF_unique)

# Create a DF for all Patients with BA.best
IN.7.unique <- bind_rows(IN.7.list.unique) %>%
  mutate(allele.best=paste0(allele.best.BA.MHCflurry, sep=";", allele.
```

```
      best.BA.netMHC, sep=";", allele.best.rank.MHCflurry, sep=";",
      allele.best.rank.netMHC))
394 IN.7.unique["allele.best"] <- unlist(lapply(IN.7.unique$allele.best,
    function(x) {paste0(unique(as.vector(str_split(x,";", simplify = T)
    )),collapse=", ")} ))
395 IN.7 <- bind_rows(IN.7.list)
396
397 # Create a DF for each Patient
398 for(i in names(IN.6.list.unique)){
399  assign(paste0(i), IN.6.list.unique[[i]])
400 }
401
402 #### IN.8: Create a DF with all peptides (also non-predicted ones and "
    non-uniques")
403 # IN.7.left <- IN.1.all %>%
404 # filter(!(Seq %in% IN.7.unique$Seq)) %>%
405 # mutate(TumorVF.StrelkaRNA = TumorAD.StrelkaRNA / (TumorAD.StrelkaRNA+
    TumorRD.StrelkaRNA)) %>%
406 # mutate(TumorVF.Mutect2 = TumorAD.Mutect2 / (TumorAD.Mutect2+TumorRD.
    Mutect2))
407 # colnames.minimal <- intersect(colnames(IN.7.left), colnames(IN.7.
    unique))
408 # colnames.new <- setdiff(colnames(IN.7.unique),colnames(IN.7.left))
409 # IN.7.left <- IN.7.left %>%
410 # select(colnames.minimal)
411 # IN.7.left[c(colnames.new)] <- NA
412 # IN.8 <- rbind(IN.7.unique, IN.7.left)
413 #
414
415 #### BLAST (IN.9) ####
416
417 ## Export to Fasta for BLAST!!!!!
418
419 # add a patient-peptide specific ID
420 IN.7.Order <- IN.7.unique %>%
421  group_by(Patient_ID) %>%
422  #mutate(Seq_ID=letters[row_number()]) %>%
423  mutate(Seq_ID=sprintf("%03d", row_number())) %>%
424  mutate(Seq_ID=paste0(Patient_ID,sep="_",Seq_ID,sep="_",Seq)) %>%
425  select(Seq_ID,everything())
426
427 IN.7.FASTA <- IN.7.Order %>%
428  mutate(Seq_ID=paste0(sep=">",Seq_ID)) %>%
429  mutate(Seq=as.character(Seq)) %>%
430  ungroup() %>%
431  select(Seq_ID,Seq) %>%
```

```
  as.data.frame()

temp <- data.frame(x=1:(2*nrow(IN.7.FASTA)))
for (x in temp$x) {
  temp[x,2] <- c(IN.7.FASTA[floor((x-1)/2)+1,((x+1)%%2)+1])
}
IN.7.FASTA <- temp %>%
  select(-x)

write_delim(IN.7.FASTA, file = "Peptides/blast/FASTA_for_blast/IN.7.
    FASTA.csv", delim = "", col_names = F)

############ INTERMEDIATE STEP: BLAST WITH NCBI, DOWNLOAD "HIT TABLE
    CSV" AND REPLACE EXISTING FILE IN FOLDER BLAST_RESULT
    ##############

## Import blast Hit Table ##
path.blast.table = list.files(path = "Peptides/blast/blast_result/")
BLAST.hits <- read_csv(file = paste0("Peptides/blast/blast_result/",
    path.blast.table), col_names = F)
colnames(BLAST.hits) <- c("Seq_ID", "locus", "percent_identity", "n_pep
    _frame", "n_missmatch", "n_gap_loci", "aa_start", "aa_end", "tsc_aa
    _start", "tsc_aa_end", "e_value", "max_score", "positives")
BLAST.hits <- BLAST.hits %>%
  mutate(Seq=str_replace_all(Seq_ID, c("IN_"="", "Me_"="", "
      [1234567890]"="", "_"=""))) %>%
  mutate(n_pep=str_length(Seq)) %>%
  select(Seq_ID,Seq,n_pep, everything()) %>%
  filter(n_pep_frame>=n_pep) %>%
  filter(percent_identity==100)
BLAST.hits.summary <- BLAST.hits %>%
  distinct(Seq,locus,tsc_aa_start, .keep_all = T) %>%
  group_by(Seq_ID) %>%
  summarise(n_identicalhits=n(), hit_loci=paste0(locus, collapse = ";")
      , e_value=mean(e_value), Seq=first(Seq)) %>%
  select(Seq,everything(),-Seq_ID)

## Create DF with BlAST Information and kick-out all entries with more
    than 2 BLAST-hits ##
IN.9 <- merge(IN.7.Order, BLAST.hits.summary, all = T) %>%
  rename("BLAST.n_identicalhits"=n_identicalhits, "BLAST.hit_loci"=hit_
      loci, "BLAST.e_value"=e_value) %>%
  filter(BLAST.n_identicalhits <=2 | is.na(BLAST.n_identicalhits)) %>%
  select(Patient_ID, Master_ID_group, Metastasis, Seq, everything())
      %>%
  arrange(Patient_ID)
```

```
467
468 ###### go on only for Data with Patient-wise shared peptides #######
469
470 ## Every entry unique in Seq
471 IN.9.Order.1 <- IN.9 %>%
472  select(-Patient_ID,-(Tumor_entity:Tumor_origin),-BLAST.n_
         identicalhits,-BLAST.e_value,-BA.best.MHCflurry,-BA.best.netMHC,-
         allele.best.BA.MHCflurry,-allele.best.BA.netMHC,-TumorVF.Mutect2
         ,-TumorVF.StrelkaRNA) %>%
473  group_by(Seq) %>%
474  summarise_all(list(first))
475 IN.9.Order.2 <- IN.9 %>%
476  group_by(Seq) %>%
477  summarise(Seq_ID_n=n(), Patient_ID=paste0(Patient_ID, collapse = ",")
         , Tumor_entity=paste0(Tumor_entity, collapse = ","), Tumor_entity
         _short=paste0(Tumor_entity_short, collapse = ","), Tumor_state=
         paste0(Tumor_state, collapse = ","), Metastatic_site=paste0(
         Metastatic_site, collapse = ","), Tumor_origin=paste0(Tumor_
         origin, collapse = ","), BLAST.n_identicalhits=mean(BLAST.n_
         identicalhits), BLAST.e_value=mean(BLAST.e_value), BA.best.
         MHCflurry.min=min(BA.best.MHCflurry), BA.best.MHCflurry.max=max(
         BA.best.MHCflurry), BA.best.netMHC.min=min(BA.best.netMHC), BA.
         best.netMHC.max=max(BA.best.netMHC), allele.best.BA.MHCflurry=
         paste0(allele.best.BA.MHCflurry, collapse = ";"), allele.best.BA.
         netMHC=paste0(allele.best.BA.netMHC, collapse = ";"), TumorVF.
         StrelkaRNA=mean(TumorVF.StrelkaRNA), TumorVF.Mutect2=mean(TumorVF
         .Mutect2))
478 IN.9.Order.2["allele.best.BA.MHCflurry"] <- unlist(lapply(IN.9.Order.2$
        allele.best.BA.MHCflurry, function(x) {paste0(unique(as.vector(str_
        split(x,";", simplify = T))),collapse=",")} ))
479 IN.9.Order.2["allele.best.BA.netMHC"] <- unlist(lapply(IN.9.Order.2$
        allele.best.BA.netMHC, function(x) {paste0(unique(as.vector(str_
        split(x,";", simplify = T))),collapse=",")} ))
480 ## Label shared Peptides with "*" and distinguish between BA.best.min
        and BA.best.max for those peptides
481 IN.9.Order <- merge(IN.9.Order.1,IN.9.Order.2) %>%
482  select(Seq_ID,Seq,Patient_ID,Master_ID_group,Metastasis,SeqMarked,
         Chrom, Pos, gene,TrueHit,MS.tool,Tumor_entity:Tumor_origin,BLAST.
         n_identicalhits,BLAST.hit_loci,BLAST.e_value,everything()) %>%
483  mutate(is.crosspatient_pep = Seq_ID_n > 1,
484       Seq_ID = ifelse(is.crosspatient_pep, paste0(str_sub(Seq_ID,1,9)
             , "*", as.character(Seq), sep = ""), Seq_ID)) %>%
485  select(-is.crosspatient_pep, -Seq_ID_n) %>%
486  mutate(BA.best.MHCflurry.min=round(BA.best.MHCflurry.min, digits = 0)
         , BA.best.netMHC.min=round(BA.best.netMHC.min, digits = 0), BA.
         best.MHCflurry.max=round(BA.best.MHCflurry.max, digits = 0), BA.
```

```
      best.netMHC.max=round(BA.best.netMHC.max, digits = 0), TumorVF.
      StrelkaRNA=round(TumorVF.StrelkaRNA, digits = 3), TumorVF.Mutect2
      =round(TumorVF.Mutect2, digits = 3)) %>%
487  arrange(Seq_ID) %>%
488  select(-header, everything()) %>%
489  select(Seq_ID:calledBy, allele.best, BA.best.MHCflurry.min, BA.best.
      netMHC.min, Rank.best.MHCflurry, Rank.best.netMHC, TumorVF.
      StrelkaRNA:TumorVF.Mutect2, everything())
490 peptides.n.summary <- IN.9.Order %>%
491  group_by(Patient_ID) %>%
492  summarise(N_peptides=n())
493
494
495 ## Create DF for HLA matching LCLs (input DF is IN.9 where the list is
     not collapsed to unique Seq yet, but Blasts are already filtered
     out) ### done with MHCflurry ###
496 IN.alleles <- IN.9 %>%
497  group_by(allele.best) %>%
498  summarise(Seq_IDs=paste0(Seq_ID, collapse = ","), Patient_IDs=paste0(
      Patient_ID, collapse = ";"))
499 IN.alleles["Patient_IDs"] <- unlist(lapply(IN.alleles$Patient_IDs,
     function(x) {paste0(unique(as.vector(str_split(x,";", simplify = T)
     )),collapse=",")} ))
500  #distinct(Patient_ID,allele.best, .keep_all = T) %>%
501 IN.alleles.patients <- IN.9 %>%
502  group_by(Patient_ID, allele.best.BA.MHCflurry, allele.best.rank.
      MHCflurry) %>%
503  summarise(Seq_IDs=paste0(Seq_ID, collapse = ","), BA.min=min(BA.best.
      MHCflurry,BA.best.netMHC)) %>%
504  mutate(isequal=(allele.best.BA.MHCflurry==allele.best.rank.MHCflurry)
      , allele.best.rank.MHCflurry=ifelse(isequal, "", allele.best.rank
      .MHCflurry), BA.min=round(BA.min, digits = 0)) %>%
505  select(-isequal)
506
507 ## Create DF with WT-peptides from possible substitution-peptides
508
509 # fs: Frameshift mutation
510 # If **ABCDEF --> Mutation lies x bp upstream and leads to complete
     frameshift peptide --> no WT
511 # If ABCD*E*F --> mutation lies in the peptide and all downstream aa's
     are changed --> WT has 1 to 9 changed aa's
512
513 IN.WT <- IN.9 %>%
514  mutate(temp=as.vector(strsplit(header, split='|', fixed=T)))
515 IN.WT["substitution"] <- unlist(lapply(IN.WT$temp, function(x) {unlist(
     x)[10]}))
```

```
IN.WT <- IN.WT %>%
  select(-temp) %>%
  mutate(substitution=str_remove(substitution, pattern = "p.")) %>%
  mutate(substitution=str_replace_all(substitution, c(
    "Ala"="A", "Arg"="R", "Asn"="N", "Asp"="D", "Cys"="C",
    "Gln"="Q", "Glu"="E", "Gly"="G", "His"="H", "Ile"="I",
    "Leu"="L", "Lys"="K", "Met"="M", "Phe"="F", "Pro"="P",
    "Ser"="S", "Thr"="T", "Trp"="W", "Tyr"="Y", "Val"="V",
    "[1234567890]"=">", ">>>>"=">", ">>>"=">", ">>"=">"
  ))) %>%
  mutate(substitution=ifelse(substitution=="NA", NA, substitution)) %>%
  mutate(AA_WT=ifelse(is.na(SeqMarked), NA, str_sub(substitution, start
       = 1, end = 1))) %>%
  mutate(AA_WT=ifelse(AA_WT=="*", NA, AA_WT)) %>%
  mutate(AA_WT=ifelse(grepl("[*][*]", SeqMarked), NA, AA_WT)) %>% # AA_
      WT is "NA" for upstream mutations
  mutate(WT_Seq=str_replace(SeqMarked, "[*].[*]", AA_WT)) %>%
  mutate(WT_Seq=ifelse(grepl("[*][*]", WT_Seq), NA, WT_Seq)) %>%
  rename(Master_ID=Master_ID_group) %>% # (!!!) column renamed!
  filter(is.na(WT_Seq)==F) #%>%
  #select(SeqMarked, substitution, AA_WT, WT_Seq)
IN.WT.export <- IN.WT %>%
  select(Patient_ID, Master_ID, Seq_ID, WT_Seq)


#### Export peptide-list for mhc-flurry
IN.WT.exported <- process_df_for_prediction_all(IN.WT.export, Seq_
    column = "WT_Seq", reference.HLA = reference.HLA)
  #purrrlyr::by_row(~write.csv(.$data, file = paste0("../../Python/
      input_files/wt_peptides/", .$id, ".csv")))

for (i in unique(IN.WT.exported$Patient_ID)){
  IN.WT.exported.selected <- IN.WT.exported %>%
    filter(Patient_ID==i)
  write.csv(IN.WT.exported.selected, file = paste0("../../Python/input_
      files/wt_peptides/", i, ".csv"), row.names=FALSE)
}

#### Import WT-predictions as DF ####
path.predictions.wt <- dir('../../Python/result_files/wt_peptides/
    models_class1_presentation/', pattern="*.csv", full.names=TRUE)
IN.WT.predictions.mhcflurry.long <- lapply(path.predictions.wt, read.
    csv)

process.predictions_WT <- function(x)
{
```

```
555   select(x, Patient_ID, Master_ID, Seq_ID, allele, peptide, mhcflurry_
          affinity, mhcflurry_affinity_percentile) %>%
556     group_by(Patient_ID, Master_ID, Seq_ID, allele, peptide) %>%
557     summarise(BA.prediction.mhcflurry=mean(mhcflurry_affinity), BA.
          percentile.mhcflurry=mean(mhcflurry_affinity_percentile)) %>%
558     gather(key=mhcflurry_type, value=prediction.value, -(Patient_ID:
          peptide)) %>%
559     unite(temp, allele, mhcflurry_type, sep = ".") %>%
560     spread(temp, prediction.value) %>%
561     rename(Seq="peptide")
562 }
563 sum_up_DF_WT <- function(x) {
564   temp <- select(x, Patient_ID, Master_ID, Seq_ID, substitution, WT_Seq
          , Chrom, Pos, gene, BA.best, allele.BA.best, BA.rank.best, allele
          .rank.best, header)
565 }
566
567 # long to wide
568 IN.WT.predictions.mhcflurry.wide <- lapply(IN.WT.predictions.mhcflurry.
      long, process.predictions_WT)
569 # as list
570 IN.WT.list <- split(IN.WT, f=IN.WT$Patient_ID)
571 # Merge with predictions
572 IN.WT.BA.list <- purrr::map2(IN.WT.list, IN.WT.predictions.mhcflurry.
      wide, merge, by.x=c("Seq_ID", "Master_ID"), by.y=c("Seq_ID", "
      Master_ID")) #by.x=c("Seq","Master_ID"), by.y=c("Seq","Master_ID")
573 # find best binders
574 IN.WT.BA.list <- lapply(IN.WT.BA.list, find_best_binder)
575 IN.WT.BA.list <- lapply(IN.WT.BA.list, sum_up_DF_WT)
576 IN.WT.BA <- bind_rows(IN.WT.BA.list) %>%
577   mutate(allele.best=paste0(allele.BA.best, sep=";", allele.rank.best))
          %>%
578   select(Patient_ID:gene, allele.best, everything()) %>%
579   mutate_at(c("BA.best", "BA.rank.best"), round, digits=6)
580 IN.WT.BA["allele.best"] <- unlist(lapply(IN.WT.BA$allele.best, function
      (x) {paste0(unique(as.vector(str_split(x,";", simplify = T))),
      collapse=", ")} ))
581
582 ###### Check which peptides have significantly improved BA/rank through
       new HLA-C predictions with MHCflurry and netMHC ########
583
584 check.HLA.C.binder.BA <- function(x){
585   temp <- x %>%
586     mutate_if(is.numeric, round, digits=2) %>%
587     filter( (grepl("C", allele.best.BA.MHCflurry) & BA.best.MHCflurry
          <500 ) | (grepl("C", allele.best.BA.netMHC) & BA.best.netMHC
```

```
       <500)) %>%
588    filter_at(vars(starts_with("HLA-A"), starts_with("HLA-B"), -contains
           ("rank")), all_vars(.>2000) )
589 }
590
591 check.HLA.C.binder.rank <- function(x){
592   temp <- x %>%
593     mutate_if(is.numeric, round, digits=2) %>%
594     filter( (grepl("C", allele.best.rank.MHCflurry) & Rank.best.
           MHCflurry<2 ) | (grepl("C", allele.best.rank.netMHC) & Rank.best
           .netMHC<2)) %>%
595     filter_at(vars(starts_with("HLA-A"), starts_with("HLA-B"), -contains
           ("prediction")), all_vars(.>2) )
596 }
597
598 remove.empty.lists <- function(x){
599   if(nrow(x)==0){return(NA)}
600   else
601     return(x)
602 }
603
604 IN.7.NEW.HLA.C.WB.BA.1 <- lapply(IN.6.list, check.HLA.C.binder.BA)
605 IN.7.NEW.HLA.C.WB.BA.2 <- lapply(IN.7.NEW.HLA.C.WB.BA.1, sum_up_DF)
606 IN.7.NEW.HLA.C.WB.BA.3 <- lapply(IN.7.NEW.HLA.C.WB.BA.2, modify_DF_
       unique)
607 IN.7.NEW.HLA.C.WB.BA <- bind_rows(IN.7.NEW.HLA.C.WB.BA.3)
608
609 IN.7.NEW.HLA.C.WB.rank.1 <- lapply(IN.6.list, check.HLA.C.binder.rank)
610 IN.7.NEW.HLA.C.WB.rank.2 <- lapply(IN.7.NEW.HLA.C.WB.rank.1, sum_up_DF)
611 IN.7.NEW.HLA.C.WB.rank.3 <- lapply(IN.7.NEW.HLA.C.WB.rank.2, modify_DF_
       unique)
612 IN.7.NEW.HLA.C.WB.rank <- bind_rows(IN.7.NEW.HLA.C.WB.rank.3)
613
614 #
       ##########################################################################
615
616 #### Kick-Out Mel_15 peptides #####
617 IN.7.Mel15 <- IN.7.unique
618 IN.7 <- IN.7.unique %>%
619   filter(Patient_ID!="Me_15")
620 IN.10.Mel15 <- IN.9
621 IN.10 <- IN.9 %>%
622   filter(Patient_ID!="Me_15")
623 IN.WT.Mel15 <- IN.WT.BA
624 IN.WT <- IN.WT.Mel15 %>%
```

```
625   filter(Patient_ID!="Me_15")
626
627 ##### Which DFs are needed??? All functions + some DFs
628 rm(list=setdiff(setdiff(ls(), lsf.str()), c(
629                                       "IN.7.Mel15",
630                                       "IN.7",
631                                       "IN.9.Order",
632                                       "IN.10",
633                                       "IN.10.Mel15",
634                                       "IN.WT.Mel15",
635                                       "IN.WT",
636                                       "IN.8",
637                                       "IN.6.list",
638                                       "IN.6.list.unique",
639                                       "IN.7.NEW.HLA.C.WB.BA",
640                                       "IN.7.NEW.HLA.C.WB.rank",
641                                       "IN.alleles",
642                                       "IN.alleles.patients",
643                                       "predictions.all")))
644
645 #### KICK-OUT PEPTIDES #####
646 # delete row 5, 6, 19 and 40
647 # IN.10 <- IN.9[-c(5,6,19,40), ]
648
649 #### EXPORT to .csv ####
650 export.data <- function(){
651
652   # List IN.6 - Unique
653   erer::write.list(IN.6.list.unique, file = "Peptides/exported_csv/IN.
        peptides.list.unique.csv")
654   write.xlsx(IN.6.list.unique, file = "Peptides/exported_xlsx/IN.
        peptides.list.unique.xlsx")
655   # DF IN.7 - Unique
656   write_csv(IN.7.unique, path = "Peptides/exported_csv/IN.peptides.
        summary.unique.csv")
657   write.xlsx(IN.7.unique, file = "Peptides/exported_xlsx/IN.peptides.
        summary.unique.xlsx")
658   # DF IN.9
659   write_csv(IN.9, path = "Peptides/exported_csv/IN.9.csv")
660   write.xlsx(IN.9, file = "Peptides/exported_xlsx/IN.9.xlsx")
661   # DF IN.10
662   #write_csv(IN.10, path = "Peptides/exported_csv/IN.10.csv")
663   #write.xlsx(IN.10, file = "Peptides/exported_xlsx/IN.10.xlsx")
664   # DF IN.WT.BA
665   write_csv(IN.WT.BA, path = "Peptides/exported_csv/IN.WT.BA.csv")
666   write.xlsx(IN.WT.BA, file = "Peptides/exported_xlsx/IN.WT.BA.xlsx")
```

```
667   # DF IN.alleles
668   write_csv(IN.alleles, path = "Peptides/exported_csv/IN.alleles.csv")
669   write.xlsx(IN.alleles, file = "Peptides/exported_xlsx/IN.alleles.xlsx
          ")
670   # DF IN.alleles.patients
671   write_csv(IN.alleles.patients, path = "Peptides/exported_csv/IN.
          alleles.patients.csv")
672   write.xlsx(IN.alleles.patients, file = "Peptides/exported_xlsx/IN.
          alleles.patients.xlsx")
673   # DF IN.NEW.HLA.C.WB.BA
674   write_csv(IN.7.NEW.HLA.C.WB.BA, path = "Peptides/exported_csv/IN.
          improve.BA.by.HLA_C.csv")
675   write.xlsx(IN.7.NEW.HLA.C.WB.BA, file = "Peptides/exported_xlsx/IN.
          improve.BA.by.HLA_C.xlsx")
676   # DF IN.NEW.HLA.C.WB.rank
677   write_csv(IN.7.NEW.HLA.C.WB.rank, path = "Peptides/exported_csv/IN.
          improve.rank.by.HLA_C.csv")
678   write.xlsx(IN.7.NEW.HLA.C.WB.rank, file = "Peptides/exported_xlsx/IN.
          improve.rank.by.HLA_C.xlsx")
679
680 }
```

Listing 5: Pipeline for assessment of specifications of neoantigen candidates

2.4 Import reference data

```
1  ############ IMPORT REFERENCES #############
2  # for the raw data please contact p.seifert@tum.de
3
4  library(tidyverse)
5  library(openxlsx)
6  library(stringr)
7
8  reference.entity <- read.csv2("rawfiles/references/entity_reference_
       masterID_2.csv")
9  reference.master <- read_csv2("rawfiles/references/master_reference.csv
       ")
10 reference.HLA <- read_csv2("rawfiles/references/HLA.types_reference_2.
       csv")
11 reference.HLA.thesis <- read_csv2("rawfiles/references/HLA.types_
       reference_2_thesis.csv")
12 reference.alleles.available <- read_csv2("rawfiles/references/available
       .alleles_reference_3.csv")
```

```
reference.alleles.available.netMHC <- read_csv2("rawfiles/references/
    available.alleles_reference_netMHC.csv")
reference.allele.frequency <- read_csv2("rawfiles/references/allele_
    frequency.csv") %>%
 mutate(allele_frequency=as.numeric(allele_frequency))
reference.genelength <- read_tsv("rawfiles/references/hsa_GRCh38_
    gencode_v36_genes.tsv")
reference.genelength2 <- read_tsv("rawfiles/references/hsa_GRCh38_
    gencode_v36_transcripts.tsv")

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 = 14),
   axis.text.x = element_text(angle = 0),
   axis.title = element_text(size = 16,face="bold"),
   legend.text = element_text(size = 16),
   legend.title = element_text(size= 16,face="bold")
 )
}
```

Listing 6: Pipeline for loading all required reference data

Appendix B: Raw data

1 Entities of ImmuNeo patients

Patient	Tumor entity	Tumor entity group	Tumor origin
01	Thymus-CA	Carcinoma	Thymus
02	Mamma-CA	Carcinoma	Breast
03	Desmoplas. small round cell T.	Sarcoma	Abdomen
04	Renal Cell CA	Carcinoma	Kidney
05	Leiomyosarcoma	Sarcoma	Muscle
08	Neuroendocrine Ovarian-CA	Carcinoma	Ovar
09	Thyroid-CA	Carcinoma	Thyroid
11	Endometrium-CA	Carcinoma	Endometrium
13	Nonseminomatous Germ Cell T.	Other	Germ Cells
14	Melanoma	Melanoma	Skin
15	Testicle-CA	Carcinoma	Testicle
16	Adenocarcinoma	Carcinoma	Salivary Gland
17	Melanoma	Melanoma	Skin
18	Mamma-CA	Carcinoma	Breast
19	Melanoma	Melanoma	Skin
20	Testicle-CA	Carcinoma	Testicle
22	Melanoma	Melanoma	Skin
23	Rhabdomyosarcoma	Sarcoma	Muscle
24	Adrenocortical-CA	Carcinoma	Kidney
25	WT-GIST	Other	Intestine
26	Mucoepidermoid-CA	Carcinoma	Salivary Gland

Patient	Tumor entity	Tumor entity group	Tumor origin
27	Epitheloid Fibrosarcoma	Sarcoma	Connective Tissue
28	Clear Cell Sarcoma	Sarcoma	Soft tissue
30	Synovial Sarcoma	Sarcoma	Soft tissue
31	Rhabdomyosarcoma	Sarcoma	Muscle
32	Osteosarcoma	Sarcoma	Bone
33	Atypical Carcinoid of the Lung	Carcinoma	Lung
34	Mucinous Adenocarcinoma	Carcinoma	NA
35	Fibromyxoid Sarcoma	Sarcoma	NA
36	Adenocarcinoma	Carcinoma	Gastroesophageal Junction
37	Appednix-CA	Carcinoma	Appendix
38	MPNST	Sarcoma	Connective Tissue

2 List of NACs

2.1 Actual

Seq ID	Seq	Gene	MS tool	Rank
01_001	ALSGHLETL	ANXA2P2	PFIND + PROSIT	0.0486
01_002	GHPSGARAM	RAB8A	PFIND	0.5538
01_003	KELCKQIQL	GARS	PROSIT	0.2636
01_004	KGDSPQVKLKY	TFAM	PFIND + PROSIT	3.5168
01_005	VEDHRARDVEV	AC018630.2	PFIND + PROSIT	3.2747
01_006	VTGAVVSAVMCRK	HLA-K	PFIND	27.0765
02_001	TGGQKYRTK	ZFAND5	PFIND	2.781
03_001	AASASRVQVI	SNHG4	PFIND	1.5356
03_002	ESKDFCVM	TBCA	PFIND	12.4599
03_003	GSHDQAMHF	GPSM2	PFIND + PROSIT	1.2015
03_004	TDGGGRAKL	ARL8B	PFIND + PROSIT	0.6051
03_005	TFQKKTKEM	FRG1KP	PFIND + PROSIT	0.0531

Seq ID	Seq	Gene	MS tool	Rank
03_006	VDSRGSLF	WASHC2A	PROSIT	0.4512
04_001	AGVVLGGL	PSMD8	PFIND	13.3492
04_002	FLLLLLKNF	SF3B1	PFIND	1.0337
04_003	GHGQPWNSL	LRP1	PFIND	0.1359
04_004	GLAATFASL	MTMR9LP	PFIND	0.8625
04_005	HAGAALHLH	ZBTB12	PFIND	2.0799
04_006	IQDGSIHRI	SDHAP1	PFIND + PROSIT	0.0432
04_007	KLQNASKKLF	L3MBTL4	PFIND	0.7299
04_008	KSAGIAGL	AC145207.5	PFIND	5.1398
04_009	KTKEMSNNVK	FRG1DP	PFIND	8.002
04_010	LGGTGASF	AC112491.1	PFIND	3.5645
04_011	NTLMSLSDM	MAP4K5	PFIND	4.8385
04_012	SYLSNISY	ASPH	PFIND + PROSIT	2.7266
04_013	TSLAANTF	BTF3P10	PFIND	2.3298
04_014	TVHSTSIAF	TMSB4XP4	PFIND + PROSIT	0.0561
05_002	DLLEPGGQR	AC011447.6	PROSIT	0.0315
05_003	ETNKSLLKR	CR381653.1	PFIND + PROSIT	0.0096
05_004	SLGAGRWRL	AC053513.1	PFIND	3.8333
08_001	APVLKSAR	HLA-J	PFIND	3.7673
08_002	GLEPGKCSP	TMEM161A	PFIND	13.4719
08_003	GPLGPRGSI	COL5A2	PFIND	0.0179
08_004	LSELDVSVR	NUDC	PFIND	4.9281
08_005	NRITEVSAK	OTUD6B-AS1	PFIND + PROSIT	0.2124
08_006	PQESAPAAL	CRIM1	PFIND	1.4491
08_007	SAGAAAQGRAGGAP	GAB2	PROSIT	1.65
08_008	TQALVLAPTQ	EIF4A1P4	PFIND	25.701
11_001	GGITAVTLN	KRT8P33	PFIND + PROSIT	14.6679
11_002	RGISWRSHL	SCART1	PFIND	1.3926
11_003	SAAELHHV	CALM3	PFIND	0.2549
11_004	SRSVAQAGVQR	AP003692.1	PROSIT	0.3479

Seq ID	Seq	Gene	MS tool	Rank
11_005	VAAGPGAV	PAXBP1	PFIND	1.335
13_001	KLPTLPKKY	ANAPC4	PFIND	1.9233
13_002	LFKNLTIL	MMADHC	PROSIT	1.1467
15_001	ICTTSVSK	ZDHHC20P4	PFIND	5.6149
15_002	LRAVTLIAK	AC012435.1	PFIND	3.0226
17_001	AGLSHHAL	AC145207.5	PFIND	0.5539
17_002	MQSRLTAA	AC118344.2	PFIND + PROSIT	0.5945
18_001	KTSKAKNTK	DIAPH1	PFIND	0.1559
18_002	MRLWSQLL	AC090114.2	PFIND	2.1848
19_001	GRPGTRPAL	BICDL1	PFIND	1.84
19_002	GSLNGGKPFLQAFY	ALDH1A2	PROSIT	3.1525
19_003	KKYWVGAKL	AP002840.2	PROSIT	8.9384
19_004	KVGSLAGF	CNNM1	PFIND	14.4536
19_005	MPEHQSTAL	C2;AL645922.1	PFIND + PROSIT	0.0037
19_006	RRLQRDKIA	"NARF"	PFIND	19.0906
19_007	SESNVDRLM	COPG2	PFIND + PROSIT	0.108
19_008	STLVLDEFKR	AC008038.1	PFIND + PROSIT	6.4807
19_009	VASISLTK	RPL36AL	PFIND + PROSIT	4.2156
22_001	PPSEAQPLP	SPATA5L1	PFIND	18.796
23_001	ASASQSAGIIGMSH	AC024075.2	PFIND	7.4174
23_002	GAPAPVMVEK	COL6A3	PROSIT	0.4014
24_001	LPIYGRAR	EPHB1	PROSIT	16.7907
24_002	SRVVGITGVP	SCAND2P	PFIND	7.024
24_003	STMVKGRQTTTK	LDHB	PROSIT	1.401
27_001	EGVAGPHSR	SUSD3	PFIND	0.1164
28_001	DTAPSGESR	APOPT1;AL139300.1	PFIND + PROSIT	2.912
28_002	EPLTTREI	NET1	PFIND	1.7411
28_003	GARLSSGRL	EIF3G	PFIND	1.2322
28_004	RVWDVSGLRKK	COPA	PFIND	0.0686
28_005	SPRQPPLLL	CDK13	PFIND + PROSIT	0.0011

Seq ID	Seq	Gene	MS tool	Rank
28_006	VGSGLGPGWVM	EPS8L2	PFIND	2.8371
28_007	VIHPPRPPK	PINK1-AS	PFIND + PROSIT	0.0061
30_001	QCKRSSSSYR	ZDHHC13	PFIND + PROSIT	19.721
32_001	APKSSSGFSL	AL033519.2	PFIND + PROSIT	0.0087
32_002	GPGSIQKR	LRRC37A	PFIND + PROSIT	4.6674
32_004	STMSALPNSR	AC084809.1	PFIND + PROSIT	0.0725
33_001	EAEVEESLGLR	LINC00884	PFIND	1.4252
34_001	SEVQDRAVP	ZFP36L2	PFIND + PROSIT	0.9796
36_001	AGLGGVKL	ARF5	PFIND	5.3217
37_001	ATERKEAK	SMC1A	PFIND	8.4507
37_002	DVVVVHRRR	ACAA1	PFIND + PROSIT	0.2537
37_003	GSPSLSQR	PYGO2	PFIND	4.3675
37_004	KFAQKVLR	RPL7P9	PROSIT	8.9215
37_005	RLANTQAKKAK	CDC5L	PFIND	0.2839
37_006	SAADVVVVHR	ACAA1	PFIND + PROSIT	0.0261
37_007	TVGVPTVLEKLQK	EHHADH	PFIND	6.161
37_008	VDANRKIY	TSG101	PROSIT	1.5454
38_001	DVIRKALQY	DPYD	PFIND + PROSIT	0.0021
38_002	RPHVGIHL	POFUT1	PFIND + PROSIT	0.114
38_003	SITPGTVL	RPL6;AC115223.1	PFIND	0.3868
38_004	SQSTTASLFKK	PRUNE1	PFIND + PROSIT	0.3612
38_005	STTASLFKK	PRUNE1	PFIND + PROSIT	9e-4

2.2 Deprecated

Seq ID	Seq	Gene	MS tool	Rank
01_R01	ALAAVVTEV	NA	NA	NA
01_R02	FLAKKPSAV	AL591846.1	PROSIT + pFind	0.1726
01_R04	DAAGRNSW	AL133216.2	PROSIT + pFind	10.6157

Seq ID	Seq	Gene	MS tool	Rank
01_R05	GMGSESKASF	IGFN1	pFind	8.9717
01_R06	KKGGLIGS	LINC01694	PROSIT + pFind	46.7781
01_R07	LEAKGQAL	CROCC	pFind	2.9272
01_R08	LEHGGAIMA	AC106820.2	pFind	5.7191
01_R09	LLGSAVHE	TANGO6	pFind + PROSIT	25.9041
01_R10	SHLYSDPG	TYK2	PROSIT	36.8145
01_R11	DAARRNSW	NA	NA	NA
04_R01	AGPGNRVL	PSMD8	pFind	25.2304
04_R02	CVYKNPVI	PTPN14	pFind	10.8594
04_R03	FFTLISVSF	ANAPC4	pFind	0.8044
04_R04	FLALFWITI	AP000766.1	pFind	1.192
04_R05	GAGALLCTHL	DUX4L50	pFind	9.21
04_R06	GSPGGPVSI	COL3A1	PROSIT	2.9413
04_R07	HVGGAGLEHL	AC087190.3	pFind	3.8001
04_R08	QKRLYYQLFFNCSWY	SHPRH	PROSIT	33.058
04_R09	SLPQNLLYL	AC112721.2	PROSIT	6.5878
04_R10	THIDAGRF	CXorf36	pFind	3.0535
04_R11	GLTATFASL	NA	NA	1.6
09_R01	YHLMPFRQHCWQSL	KLHL14	PROSIT	22.3645
11_R01	AAAAPARGL	SF1	pFind	19.705
11_R02	ARETLLETL	SDC4	pFind	1.4147
11_R03	ETSAPASSL	C3	pFind	31.454
11_R04	GTPSSTTL	DSCAML1	pFind	22.6244
11_R05	ISAAELHHV	CALM3	PROSIT	2.6362
11_R06	LNITHGILY	SMC4	pFind	4.095
11_R07	LNLREKKNK	TRANK1	pFind	5.158
11_R08	RLQDAVPV	ASMTL	pFind	0.4269
11_R09	SRAAAAPAR	SF1	PROSIT	1.451
19_R01	DQATCLRSTKFTIY	F9	PROSIT	8.5661
19_R02	FFQDKAWFY	PARP14	pFind + PROSIT	0.0489

Seq ID	Seq	Gene	MS tool	Rank
19_R03	GWGVAGTM	AC112721.2	pFind	18.8456
19_R04	ITRGQEFE	AC008771.1	pFind	28.9686
19_R05	LLEAGRLR	GABPB1-AS1	pFind + PROSIT	18.4337
19_R06	PTDAELMS	PYGL	pFind	0.6449
19_R07	TLGGWGGQDLR	ZNF37BP	pFind	36.865
19_R08	TNLGFSKK	PIK3C2G	PROSIT	56.5224
S*_R01	VVHVSTSQK	AC115837.1	pFind	0.1283
S*_R02	QLRASGQLK	PTMAP5	pFind	0.1844

3 Results of acDC assays (detailed)

3.1 For actual NACs

3.1.1 Not reactive

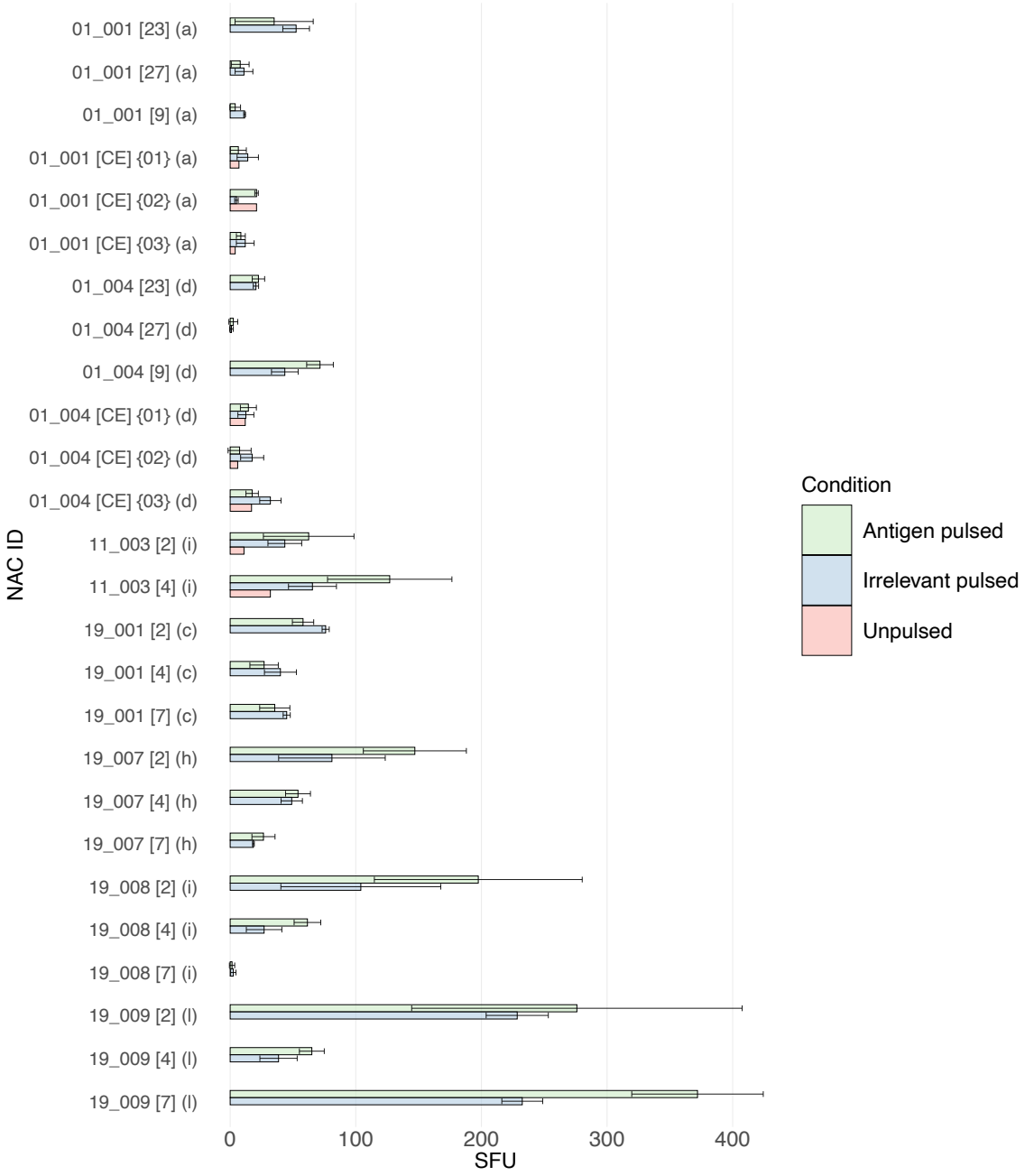


Figure 1: Detailed result of acDC assays (actual, not reactive)

3.2 For deprecated NACs

3.2.1 Reactive

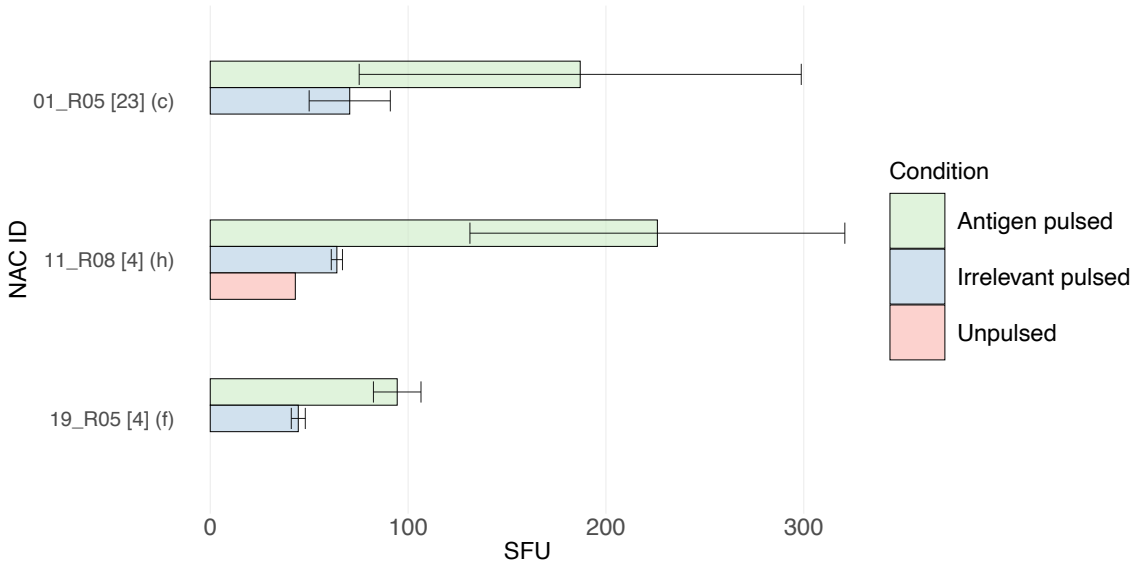


Figure 2: Detailed result of acDC assays (deprecated, reactive)

3.2.2 Not reactive

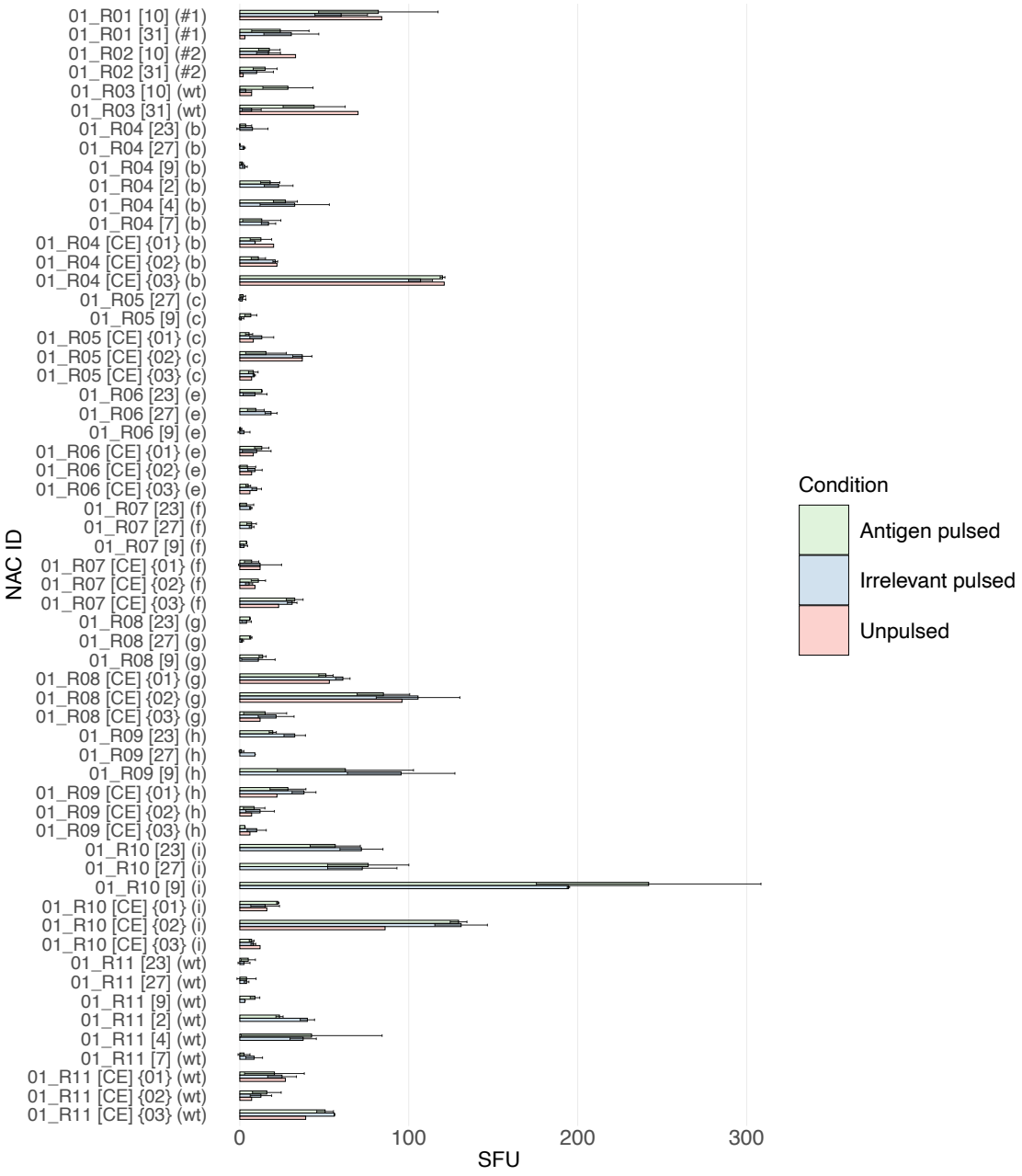


Figure 3: Detailed result of acDC assays (deprecated, reactive, 01)

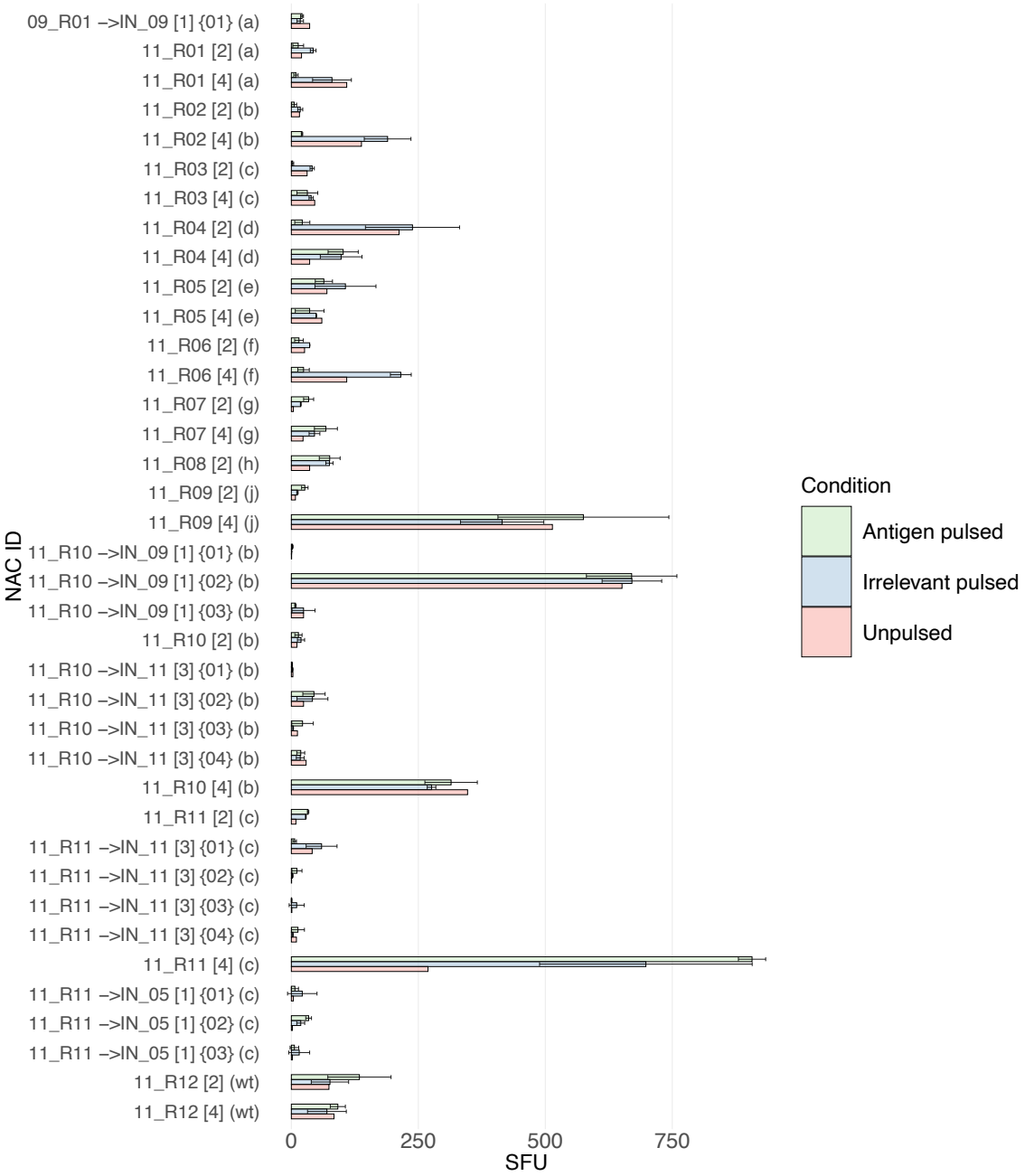


Figure 4: Detailed result of acDC assays (deprecated, reactive, 02)

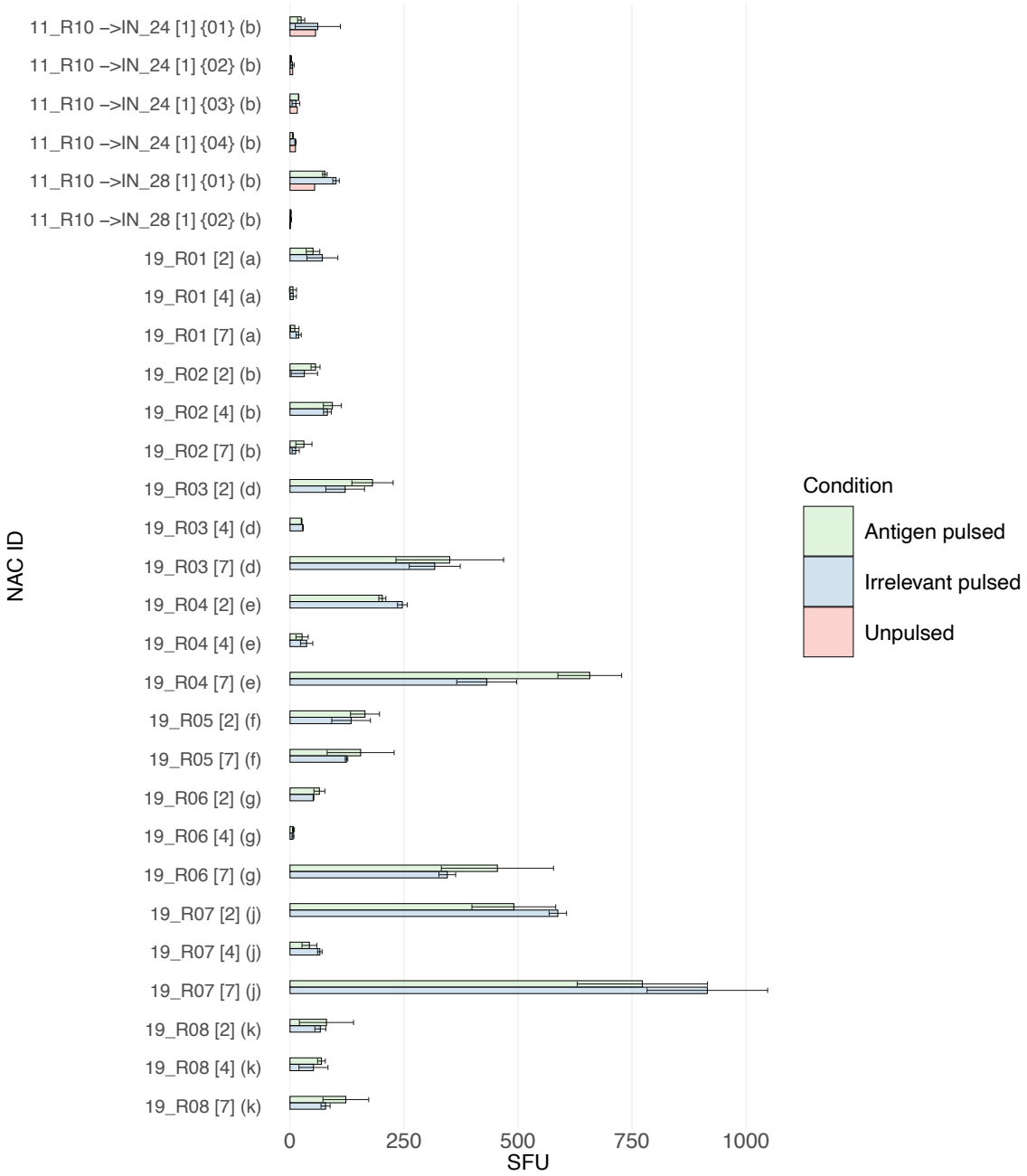


Figure 5: Detailed result of acDC assays (deprecated, reactive, 03)

Appendix C: Classification of Variants

Variant type	Intron inclu-sion	Position of mutation: peptide	Position of mutation: rel.	Position of peptide	Mutation effects (codon)	Frame of peptide	Peptide aberrant from wt-transcript seq	coding / non-coding	Relevance	Case
substitution= SNV	no	within peptide	-->	-->	-->	in-frame	-->	-->	yes	1
		not within peptide	DS	-->	-->	in-frame	-->	coding	x	2
								non-coding	cryptic MAP	3
						not in-frame	-->	-->	cryptic MAP	4
			US	-->	no	in-frame	-->	coding	x	5
								non-coding	cryptic MAP	6
						not in-frame	-->	-->	cryptic MAP	7
					new start	in-frame	-->	-->	x	8
						not in-frame	-->	-->	yes	9
					stop lost	-->	-->	-->	yes	10
	yes	within peptide	-->	exonic (other splice region variant)	-->	-->	-->	-->	yes	11
				full intronic (splice donor/splice acceptor/intron variant)	-->	-->	-->	-->	yes	12
				partially intronic (all splice region variants/intron variant)	-->	-->	yes	-->	yes	13
							no	-->	x	14
		not within peptide	DS	exonic	-->	in-frame	-->	coding	x	15
								non-coding	cryptic MAP	16
						not in-frame	-->	-->	cryptic MAP	17
				full intronic (splice acceptor variant/ intron variant)	-->	-->	yes	-->	yes	18
							no	-->	x	19
				partially intronic (splice acceptor variant/ intron variant)	-->	-->	yes	-->	yes	20
							no	-->	x	21
			US	exonic (all splice region variants/ intron variant)	-->	in-frame	-->	coding	x	22
								non-coding	cryptic MAP	23
						not in-frame	-->	-->	yes	24
				full intronic (splice donor variant /intron variant)	-->	-->	yes	-->	yes	25
							no	-->	x	26
				partially intronic (splice donor variant / intron variant)	-->	-->	yes	-->	yes	27
							no	-->	x	28

Figure 1: Classification of variants; substitutions

Variant type	Intron inclu-sion	Position of mutation: peptide	Position of mutation: rel.	Position of peptide	Mutation effects (codon)	Frame of peptide	Peptide aberrant from wt-transcript seq	coding / non-coding	Relevance	Case
insertion / deletions /duplications	no	within peptide	-->	-->	-->	in-frame	-->	-->	yes	29
						not in-frame	-->	-->	yes	30
		not within peptide	DS	-->	no	in-frame	-->	coding	x	31
								non-coding	cryptic MAP	32
						not in-frame	-->	-->	cryptic MAP	33
					stop lost	ORF rescue?	-->	-->	??	34
			US	-->	no	in-frame	-->	coding	x	35
								non-coding	cryptic MAP	36
						not in-frame	-->	-->	yes	37
					new start	in-frame	-->	coding	x	38
								non-coding	cryptic MAP	39
						not in-frame	-->	-->	yes	40
					stop lost	-->	-->	-->	yes	41
	yes	within peptide	-->	exonic (other splice region variant)	-->	in-frame	yes	-->	yes	42
							no	-->	x	43
						not in-frame	-->	-->	yes	44
				full intronic (splice donor/splice acceptor/ intron variant)	-->	-->	-->	-->	yes	45
				partially intronic (all splice region variants/ intron variant)	-->	-->	yes	-->	yes	46
							no	-->	x	47
		not within peptide	DS	exonic	-->	in-frame	-->	coding	x	48
								non-coding	cryptic MAP	49
						not in-frame	-->	-->	cryptic MAP	50
				full intronic (splice acceptor variant/ intron variant)	-->	-->	yes	-->	yes	51
							no	-->	x	52
				partially intronic (splice acceptor variant/ intron variant)	-->	-->	yes	-->	yes	53
							no	-->	x	54
			US	exonic (all splice region variants/ intron variant)	-->	in-frame	-->	coding	x	55
								non-coding	cryptic MAP	56
						not in-frame	-->	-->	yes	57
				full intronic (splice donor variant/ intron variant)	-->	-->	yes	-->	yes	58
							no	-->	x	59
				partially intronic (splice donor variant/ intron variant)	-->	-->	yes	-->	yes	60
							no	-->	x	61

Figure 2: Classification of variants; insertions, deletions, duplications

Appendix D: Protocols

In-vitro stimulation of T cells

Day	Date	Time	Procedure	Duration	Step	Substep	
-1	Montag 28.10.19	9:00 11:30	DC & T-Cell Stimulation		**1a**	1a 1b 2 3 4 5 6 7 8	Thaw PBMCs froom ImmuNEO#1 from 3 tome points Thaw PBMCs from healthy donor (JuTh untrans. 5Mio, 14.09.15) in 15mL-falcon with AIM-V (washing) Spin 500g, 5min Pour off medium Resuspend in 1ml AIM-V (per 20 million cells) Count cells and insert number in H7 Spin 500g, 5min and resuspend in appropriate amount of AIMV add cytokines (freshly thawed)
					1b	1 2 3	Distribute in wells (96 well plate, flat bottom): (200 µl per well) Add PBS to empty wells Incubate @ 37°C
					2	1	Dissolve peptides if needed in DMSO and prepare 1:10 dilutions in AIM V see table "Labor_Tabellen_Peptides_ImmuNEO" for calculation
0	Dienstag 29.10.19	10:15 12:00	T-Cell activation T-Cell stimulation Coat ELISpot plate	2h	**3** **4**	1 2 3 4 5 1 2 3	Make Master-Mixes of Medium (always thaw fresh cytokines when older than 2 weeks) Add MasterMixes to well plate Add Peptide to well plate Add PMA/Iono to + wells, add DMSO 1:10 in AIM-V to - wells Incubate @ 37°C Prepare Capture AB solution Add to Elispot plate (50 µl per well) Incubate @ 4°C for over night
1	Mittwoch 30.10.19		Wash ELISpot plate Prepare cells for ELISpot (and transfer them to ELISPot plate)		**5** **6**	1 2 3 1 2 3 4 5 6 7 8 9	Discard coating AB (in the hood) Wash 4x with PBS (200µl per well; leave it each time for 10 min) Block with 150 µl/well TCM for 45-60 min @ 37°C Take 100 µl supernatants in 96-well plate (round bottom) and freeze @ -20°C (Parafilm) Resuspend cells in 150 µl AIM-V Transfer cells to new 96 well-plate (round bottom, 200µl) Spin 500g, 5min Exchange AIM-V medium (100 µl) Discard Block of ELISpot plate Transfer cells to washed ELISpot plate Add PBS to empty wells Incubate @ 37°C
2	Donnerstag 31.10.19	+24h	Take cells to culture (from ELISpot plate) ELISpot development Prepare AEC solution (no longer than 15 min prior to use)		**7** **8** **9**	1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 7 8	Take 100 µl supernatants and freeze @ -20°C Resuspend cells in 100 µl RPMI Transfer cells to new 96 well-plate (round bottom) Spin 500g, 5min Exchange medium to TCM + IL7/15 (Culture medium) (take off a 120 µl and add 120 µl IL-7/15-TCM solution) Incubate @ 37°C, Cultivation for one week Wash 6x with PBS + 0,05% Tween (gut ausklopfen dazwischen) Add Antibody (100 µl/well) Incubate 2h @ RT Wash 6x with PBS + 0,05% Tween (gut ausklopfen dazwischen) Add Peroxidase complex (100 µl/well) Incubate (1-)2h @ RT (dark!!!) Prepare AEC Solution max. 15 min before use Vortex or mix Wash ELISpot plate 2x with PBS + 0,05% Tween (gut ausklopfen) Wash ELISpot plate 2x with PBS Pipett AEC Solution (100 µl/well) Incubate reaction in the dark until the positive control is visible (don't stop too late, spots need to be distinguishable Stop the reaction with abundant deionised water; remove the plastic and continue washing Leave the plate dry out then wrap it in paper and store it for a couple of days
4	Samstag 02.11.19		Addition of cytokines (Prepare for DAY 5/7)		**10**		Add IL-7/IL-15 to the cells (20µL/well) Prepare solution for day 5/7
7	Dienstag 05.11.19		Addition of cytokines (Prepare for DAY 5/7) Scan ELISPOT plate		**11**		Add IL-7/IL-15 to the cells (20µL/well) Prepare solution for day 5/7
8	Mittwoch 06.11.19		Addition of cytokines		**12**		Add IL-7/IL-15 to the cells
10	Freitag 08.11.19		Coat ELISpot plates		**14**	1 2 3	Prepare Capture AB solution Add to Elispot plate (50 µl per well) Incubate @ 4°C
12	Sonntag 10.11.19		Change Cells to bigger plate Addition of cytokines		**15a** **15b**	1 2 	Prepare 400 µl of TCM with IL-7/IL-15 per well in new 48 well plate Take 200 µl of each well (good growing) to the new well plate Add 20 µl IL-7/IL-15 (feed solution) to the cells (bad growing)

13	Montag 11.11.19		Count target cells	**Target Cells**	**17**	1	Take LCLs from Incubator
						2	spin down 500g, 5min
						3	pour off medium and resuspend in 1ml AIM-V
						4	Well plate (round bottom): 90µl Tryphanblue (0,4%) + 10µl cells
						5	count: c = (N_cells/#quadrants)*dilution*10.000 [1/ml]
						6	Fill in x Eppis for each pulsed peptide/ Eppi for wt-pulsed / Eppi for unpulsed
			(Wash Target cells)		**18**	1	Add RPMI (1-2 ml)
						2	Spin 500g, 5min
						3	Take off supernatants and discard (vacuum pump)
			Pulse Target cells		**19**	1	add respective Volume AIM-V and 1 µM peptide to pulsed cells/wt-pulsed (V_end = 200 µl for each)
						2	spin down unpulsed cells, add respective volume Aim-V
						3	Incubate @ 37 °C for 2h
			Prepare ELISpot plate		**20**	1	Tilt to remove the antibody
						2	Wash 4x with PBS (200µl per well; leave it each time for 10 min)
						3	Block with 150 µl/well TCM for 45-60 min @ 37°C
			Count T-cells	**T-Cells**	**21**	1	Prepare wellplate (round bottom) with Tryphanblue for each well in the according t-cell plate (5:1 dilution; Hood)
						2	Resuspend each well and then take 10 µl to the corresp. well
						3	count all 4 quadrants and insert in "Calc"
			Wash T-cells		**22**	1	Prepare one Eppi for each Peptide-blood-condition with 1ml TCM each
						2	Add V_cells from calc-sheet to each Epi
						3	and replace the volume with the corresponding volume of fresh TCM
						4	Spin 500g, 5min
						5	Take off supernatants and discard (vacuum pump)
							Resuspend cells of each Epi in TCM
							Put T cell plates back into incubator
			Prepare PMA/Iono		**23**	1	15 ml Falcon
						2	add TCM and PMA and Ionomycin
			Wash Target cells 2x	**Traget cells**	**24**	1	Add TCM (1-2 ml)
						2	Spin 500g, 5min
						3	Take off supernatants and discard (vacuum pump)
						4	Repeat washing steps (1-3) 1x
						5	Resuspend in TCM
			Pipett T-cells	**T cells**	**25**	1	Tilt ELISpot plate
						2	Pipett T-cells (100 µl) as designed (Vortex each EPI before)
			Pipett Target cells		**26**	1	Pipett Target cells as designed (Vortex each time)
			Pipett positve control			2	Pipett prepared PMA and Ionomycin solution (Step 23)
			Pipett negative control			3	Pipett 100µl TCM in wells as designed
						4	Incubate ELISpot plate @ 37 °C for 72h
14	Dienstag 12.11.19		Addition of cytokines			1	Add IL-7/IL-15 to well plates
16	Donnerstag 14.11.19				**28**	1	Take 100 µl supernatants and freeze @ -20°C
			ELISpot development		**29**	1	Wash 6x with PBS + 0,05% Tween (gut ausklopfen dazwischen)
						2	Add Antibody (100 µl/well)
						3	Incubate 2h @ RT
						4	Wash 6x with PBS + 0,05% Tween (gut ausklopfen dazwischen)
						5	Add Peroxidase complex (100 µl/well) (prepare in cell culture)
						6	Incubate 1-2h @ RT (dark!!!)
			Prepare AEC solution		**30**	1	Prepare AEC Solution max. 15 min before use (prepare in molecular lab)
						2	Vortex or mix
						3	Wash ELISpot plate 2x with PBS + 0,05% Tween (gut ausklopfen)
						4	Wash ELISpot plate 2x with PBS
						5	Pipett AEC Solution (100 µl/well)
						6	Incubate reaction in the dark until the positive control is visible ~2-10 min (don't stop too late, spots need to be distinguishable)
						7	Stop the reaction with abundant deionised water; remove the plastic and continue washing
						8	Leave the plate dry out then wrap it in paper and store it for a couple of days

Figure 1: Detailed protocol for acDC in vitro stimulation assay.

Appendix E: Legal aspects

Ethical vote

The study was approved by the institutional review boards (Ethics Commission of the Medical Faculty of Technical University Munich (protocol 193/17S) and Ethics Committee of the Medical Faculty of Heidelberg University (protocol S-206/2011)) and all patients provided written informed consent under these protocols. The study was conducted in accordance with the Declaration of Helsinki. Blood collection of healthy donors and the use of this material for the functional experiments in this study was approved by the Ethics Commission of theMedical Faculty of Technical UniversityMunich (protocol 521/18 SAS) and all participants provided written informed consent under this protocol.