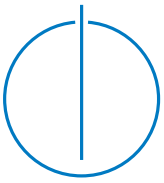


TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM School of Computation, Information and Technology

Evolution of Transmembrane Protein Prediction

Dissertation

Michael Bernhofer

Technische Universität München
TUM School of Computation, Information and Technology

Evolution of Transmembrane Protein Prediction

Michael Bernhofer

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Julien Gagneur

Prüfer*innen der Dissertation:

1. Prof. Dr. Burkhard Rost
2. Assoc. Prof. Mikael Boden Ph.D., The University of Queensland, AU

Die Dissertation wurde am 17.02.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 26.07.2023 angenommen.

Abstract

Transmembrane proteins (TMP) are essential for any living cell, facilitating several vital processes. Transporters and channel proteins regulate the internal conditions of a cell through active and passive transport of molecules. Receptor proteins receive and propagate signals across the membrane borders, enabling communication with the outside environment. However, despite their importance for molecular biology and medicine, relatively few experimentally determined structures are available. In an effort to alleviate this structure-gap, many sequence-based prediction methods for TMPs have been developed and gradually improved over the last three decades. Though they cannot completely replace 3D structures, those methods produce vital information about TMPs and their topology. As part of this thesis, we developed two prediction methods for TMPs and integrated them in web services for easy access.

Our first method, TMSEG, implements a multi-stage prediction pipeline utilizing several machine learning models. It combines two random forests (RF) with a neural network (NN) to gradually filter and improve the predictions. We carefully collected training data from the Orientations of Proteins in Membranes (OPM) database and the Protein Data Bank of Transmembrane Proteins (PDBTM). Utilizing evolutionary information in the form of position-specific scoring matrices (PSSM), TMSEG manages to perform on par with other state-of-the-art methods of its time.

We integrated TMSEG into the PredictProtein web service to enable easy access for all types of users. Running since 1992, PredictProtein is one of the oldest services for online protein structure and function prediction. Amongst others, it provides prediction methods for secondary structure, solvent accessibility, membrane proteins, conservation, protein-, RNA-, and DNA-binding, Gene Ontology (GO), and subcellular location. This diversity of information, at the click of a button, enables users to easily analyze their proteins of interest.

Once new technologies became available, we decided to improve upon our first method and developed TMbed. In contrast to TMSEG, it uses a much simpler model architecture consisting of a small convolutional neural network (CNN) coupled with a Viterbi decoder. However, the most important change was to the input features. We replaced the PSSMs with so-called embeddings produced by protein language models (pLM). Adapting the breakthroughs in natural language processing (NLP) for language models, namely the Transformer architecture, those pLMs attempt to learn the “language of life”, i.e., the inherent patterns in amino acid sequences. Often having been trained

on millions or billions of protein sequences, pLMs can produce information-rich vector representations (embeddings) for all residues in a protein sequence, which rival evolutionary information. Further, running on GPUs, pLMs can process several sequences per second, which enables pLM-based prediction methods to process whole proteomes within less than an hour. For example, TMbed can predict the over 500,000 sequences in UniProtKB/Swiss-Prot in less than nine hours, while still reaching state-of-the-art prediction performance.

Zusammenfassung

Transmembranproteine (TMP) sind für jede lebende Zelle unverzichtbar und ermöglichen mehrere lebenswichtige Prozesse. Transporter- und Kanalproteine regulieren die internen Bedingungen einer Zelle durch aktiven und passiven Transport von Molekülen. Rezeptorproteine empfangen und übertragen Signale über die Membrangrenzen hinweg, was eine Kommunikation mit der äußeren Umgebung ermöglicht. Trotz ihrer Bedeutung für die molekulare Biologie und Medizin stehen jedoch relativ wenige experimentell bestimmte Strukturen zur Verfügung. Um diese Strukturlücke zu schließen, wurden in den letzten drei Jahrzehnten viele sequenzbasierte Vorhersagemethoden für TMPs entwickelt und schrittweise verbessert. Obwohl sie 3D-Strukturen nicht vollständig ersetzen können, liefern diese Methoden wichtige Informationen über TMPs und ihre Topologie. Im Rahmen dieser Arbeit haben wir zwei Vorhersagemethoden für TMPs entwickelt und in Webdienste integriert, um einen einfachen Zugang zu ermöglichen.

Unsere erste Methode, TMSEG, implementiert eine mehrstufige Vorhersagepipeline, welche mehrere Modelle aus dem Bereich des maschinellen Lernens nutzt. Es kombiniert zwei Random Forests (RF) mit einem Neuronalen Netzwerk (NN), um die Vorhersagen schrittweise zu filtern und zu verbessern. Wir haben dafür sorgfältig Trainingsdaten aus der Datenbank Orientations of Proteins in Membranes (OPM) und der Protein Data Bank of Transmembrane Proteins (PDBTM) gesammelt. Durch die Verwendung von evolutionärer Information in Form von positionsspezifischen Scoring-Matrizen (PSSM) schafft es TMSEG, mit anderen State-of-the-Art Methoden seiner Zeit mitzuhalten.

Wir haben TMSEG in den PredictProtein Webservice integriert, um einen einfachen Zugang für alle Benutzertypen zu ermöglichen. PredictProtein ist seit 1992 in Betrieb und einer der ältesten Online-Dienste für Vorhersagen von Proteinstruktur und Funktion. Neben weiteren bietet es Vorhersagemethoden für Sekundärstruktur, freiliegende Proteinoberfläche, Membranproteine, Konservierung, Protein-, RNA- und DNA-Bindung, Gene Ontology (GO) und subzelluläre Lokalisation. Diese Vielfalt an Informationen ermöglicht es Benutzern, mit nur einem Klick Proteine mit Leichtigkeit zu analysieren.

Als neue Technologien verfügbar wurden, entschieden wir uns, unsere erste Methode zu verbessern und entwickelten TMbed. Im Gegensatz zu TMSEG verwendet es eine viel einfachere Modellarchitektur, bestehend aus einem kleinen konvolutionalen neuronalen Netzwerk (CNN) in Kombination mit einem Viterbi-Dekoder. Der wichtigste Unterschied war jedoch die Änderung der Eingabefeatures. Wir haben die PSSMs durch sogenannte Embeddings ersetzt, die von Protein-Sprachmodellen (pLM) produziert werden.

Indem sie die Durchbrüche in der natürlichen Sprachverarbeitung (NLP) für Sprachmodelle, insbesondere die Transformer-Architektur, nutzen, versuchen diese pLMs, die "Sprache des Lebens" zu erlernen, d.h. die inhärenten Muster in Aminosäuresequenzen. Oft auf Millionen oder Milliarden von Proteinsequenzen trainiert, können pLMs informationenreiche Vektorrepresentationen (Embeddings) für alle Positionen in einer Proteinsequenz generieren, die der evolutionären Information gleichwertig sind. Darüber hinaus können sie auf GPUs ausgeführt werden und mehrere Sequenzen pro Sekunde verarbeiten, was es pLM-basierten Vorhersagemethoden ermöglicht, ganze Proteome in weniger als einer Stunde zu verarbeiten. Beispielsweise kann TMbed die über 500.000 Sequenzen in UniProtKB/Swiss-Prot in weniger als neun Stunden vorhersagen, und erreicht gleichzeitig eine State-of-the-Art Vorhersagegenauigkeit.

Acknowledgements

First of all, I would like to thank Burkhard Rost for teaching and guiding me throughout my Bachelor's and Master's program, inviting me to his lab, and ultimately supervising my PhD thesis. Thank you for the long walks and talks, and the opportunities to gain experience in several areas of computational biology, medicine, and machine learning.

Thanks to Julien Gagneur and Mikael Boden for dedicating their precious time to be part of my thesis committee.

I would like to thank all of my colleagues at the Rostlab. Thank you for creating a friendly and welcoming work environment. We have had many great discussions, scientific or otherwise. I will always remember our chaotic game nights. In particular, I would like to thank Inga Weise for her help with all the administrative matters that popped up over the years, and Tim Karl for his constant efforts to keep our hardware and software running. Thanks to Maria Littmann for giving us a step-by-step guide to navigate the bureaucracy involved in writing a thesis at TUM.

Special thanks to my family, my parents and my brother, who have been supporting me throughout my entire life. Your belief in my potential has given me the strength and courage to pursue my dreams and overcome obstacles, and I will always be grateful for that. Without you, I would not be where I am today. Thanks to all my friends for offering opportunities to forget about work and enjoy all the other aspects of life. Thank you for the memories we have created, the adventures we have had, and the laughter we have shared. You have made my life richer, brighter, and more meaningful, and I cannot imagine my life without you.

Contents

List of Figures

List of Tables

Abbreviations

3D	3-Dimensional
ALM	Autoregressive Language Modeling
BLOSUM	BLOcks SUbstitution Matrix
CNN	Convolutional Neural Network
CRF	Conditional Random Field
FNN	Feed-forward Neural Network
GO	Gene Ontology
GPU	Graphical Processing Unit
GRHCRF	Grammatical-Restrained Hidden Conditional Random Field
HMM	Hidden Markov Model
LM	Language Model
LSTM	Long Short-Term Memory
MLM	Masked Language Modeling
MSA	Multiple Sequence Alignment
NLP	Natural Language Processing
NN	Neural Network
OPM	Orientations of Proteins in Membranes database
PDB	Protein Data Bank
PDBTM	Protein Data Bank of Transmembrane Proteins
pLM	Protein Language Model
PSSM	Position-Specific Scoring Matrix
pLM	Protein Language Model
RF	Random Forest
SVM	Support Vector Machine
TMB	Transmembrane Beta Strand
TMH	Transmembrane Alpha Helix
TMP	Transmembrane Protein

1. Introduction

1.1. Membrane Proteins

Countless proteins play a key part in what you could call the machinery of life, the cell. Some catalyzing chemical reactions, converting nutrients into energy-sources, others transporting molecules to their intended destinations, or receiving and sending biochemical messages. However, for those processes to function properly, the cell needs to maintain a controlled environment. For this reason, all living cells are surrounded by a plasma membrane, separating and shielding it from the outside world. By carefully regulating the internal concentrations of proteins and other molecules, the cell can control its internal conditions to fit its current needs. Within the cells of eukaryotes, there are even specialized compartments, called organelles, which are themselves surrounded by their very own membranes. However, complete isolation would ultimately result in death, either by running out of nutrients or by toxic accumulations of waste products. Thus, specialized proteins exist within the membranes to facilitate transport and communication across those barriers [1]. Channel and transporter proteins react to internal or external changes, opening or closing to enable passive or active transport of atoms and molecules. Receptor proteins, spanning both sides of a membrane, receive signals and propagate them to the other side. In other words, membrane proteins are integral parts of the logistics and communication networks of the cell.

1.1.1. Lipids and the Membrane Bilayer

Lipids are the main component of biological membranes, apart from the actual membrane proteins. Lipids are amphiphilic molecules. They consist of a hydrophilic head and one or more hydrophobic tails (Figure 1.1). When placed in water they tend to spontaneously form micelles or lipid bilayers, with their hydrophilic heads on the outside and hydrophobic tails on the inside. Biological membranes belong to the second

category, primarily containing phospholipids and glycolipids. Cholesterols are often interspersed to loosen the otherwise tightly packed structure [2], making the membrane more flexible, almost like a fluid.

1.1.2. Types of Membrane Proteins

Within those lipid bilayers sit membrane proteins. Roughly speaking, membrane proteins can be split into two groups: peripheral and integral membrane proteins. Peripheral membrane proteins are only loosely attached to the membrane, for example through hydrophobic or electrostatic interactions with the lipids or other membrane proteins. On the other hand, integral membrane proteins are embedded into the membrane. Depending on how far or often they penetrate the membrane bilayer, they are further categorized into monotopic, bitopic, and polytopic proteins. Monotopic proteins reach only partially into the membrane but do not completely cross it. Bitopic proteins cross the membrane exactly once, emerging on the other side, while polytopic proteins cross the membrane multiple times. Throughout this thesis, I will also refer to bi- and polytopic membrane proteins as transmembrane proteins (TMP). Bitopic proteins typically cross the membrane with an alpha-helical segment, while polytopic proteins come in two common shapes: tightly packed bundles of alpha helices, or barrel-like structures formed by beta strands (Figure 1.2).

1.1.3. Alpha-Helical Transmembrane Proteins

The most prevalent form of TMPs are alpha-helical TMPs. Estimates based on whole proteome predictions range from 20% to 30% [7, 8], i.e., about every fourth protein in any living organism is an alpha-helical TMP. Alpha helices are a good way to stabilize the membrane-crossing part of a TMP as they can form internal hydrogen bonds. The otherwise highly apolar environment in the hydrophobic core of the membrane bilayer makes stabilization through external bonds difficult. Amino acids found in transmembrane helices (TMH) are usually hydrophobic (e.g., A, I, L, V), interacting with the hydrophobic tails of the lipids. However, sometimes polar or charged amino acids are also part of TMHs, often shielded from the lipids by parts of the protein surface. On average, TMHs are 26 amino acids long and oriented mostly perpendicular to the membrane plane [9]. However, locally deformed membranes or tilted TMHs can allow for variations in length, resulting in TMHs with about 15 to over 40 residues [10]. Due to

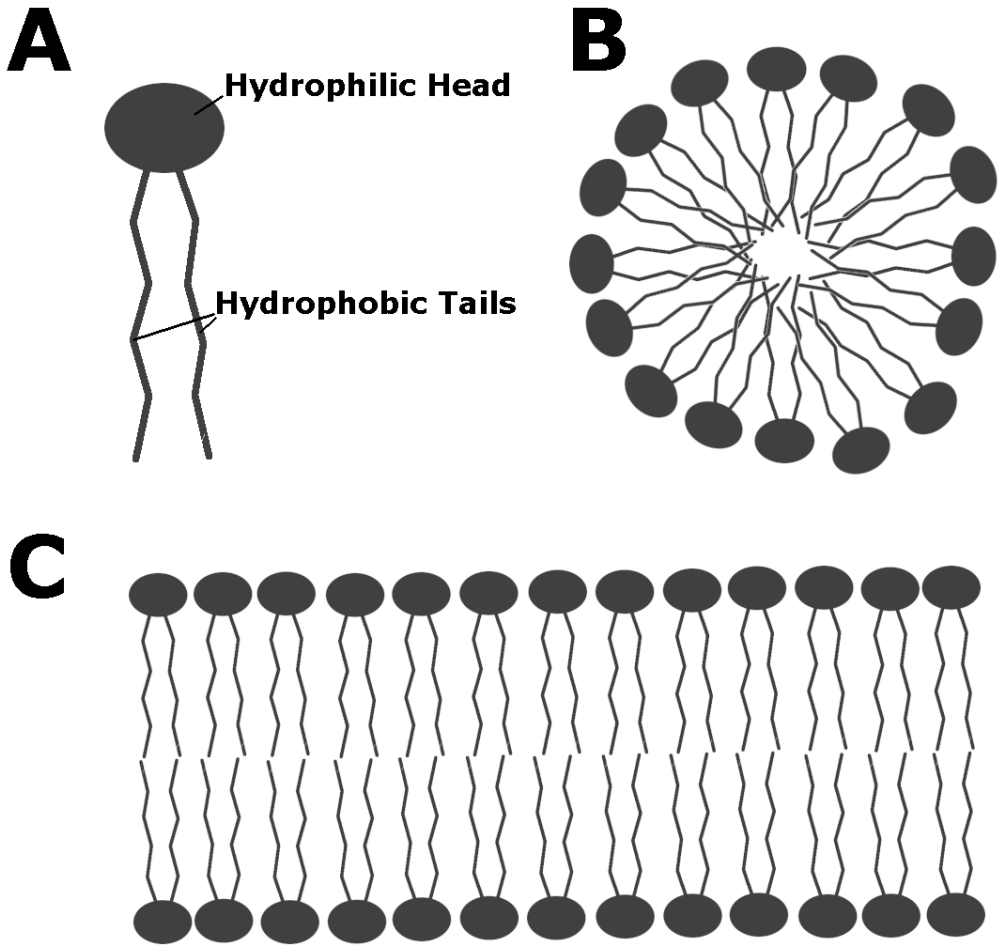


Figure 1.1.: When placed in water, lipids form micelles or bilayers, shielding their hydrophobic tails from the water while exposing their hydrophilic heads. **A:** Amphiphilic lipid with a hydrophilic head and hydrophobic tails. **B:** Lipids forming a micelle. **C:** Lipids forming a membrane bilayer.

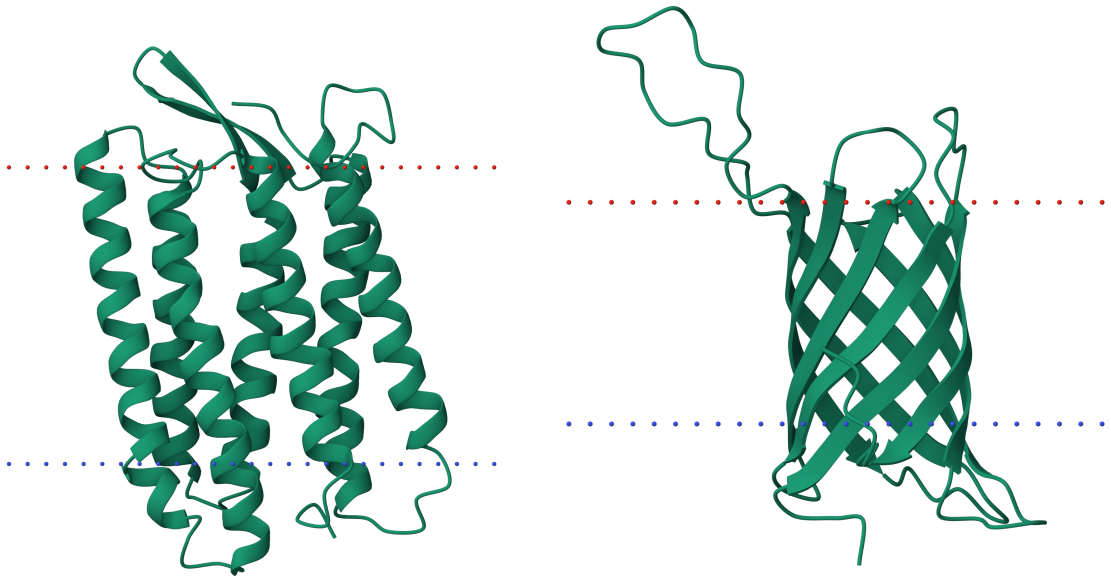


Figure 1.2.: The two main structural types of transmembrane proteins (TMP). **Left:** 3D structure (PDB: 1UAZ [3]) of the alpha-helical TMP Archaerhodopsin-1 located in the plasma membrane of *Halorubrum ezzemoulense*. **Right:** 3D structure (PDB: 2JMM [4]) of the beta-barrel outer membrane protein A (ompA) located in the outer membrane of *Escherichia coli*. Dotted lines represent the membrane boundaries annotated in OPM [5] (red: outside, blue: inside). Images created using Mol* Viewer [6].

their prevalence in nature, alpha-helical TMPs are the best-studied type of membrane proteins, and the focus of most sequence-based TMP prediction methods.

1.1.4. Beta-Barrel Transmembrane Proteins

The other and less common type of TMPs are beta-barrel TMPs. Here, the backbone hydrogen bonds are internally satisfied by a barrel-like structure made entirely of beta strands. A number of antiparallel beta strands are tilted about 45° relative to the membrane plane and arranged next to each other to form a cylindrical shape [11]. Thus, each beta strand can form bonds with the neighboring strands, with the first and last beta strand in the sequence completing the barrel. Consecutive residues in individual transmembrane beta strands (TMB) alternate between lipid-facing and pore-facing (i.e. towards the inside of the barrel). This typically results in a sequence of residues that alternate between hydrophobic and polar amino acids. Unlike alpha-helical TMPs, beta-barrel TMPs have only been found in gram-negative bacteria, making up about 3% of their proteomes [12, 13], and in the outer membranes of chloroplasts and mitochondria of eukaryotic cells.

1.1.5. The Aromatic Belt and the Positive-Inside Rule

In addition to the obvious preference of hydrophobic amino acids in the core of TMH, scientists discovered two other phenomena. First, the so-called aromatic belt or aromatic cuff [14–16], which describes the prevalence of TYR and TRP at the ends of TMHs and TMBs. Those polar and aromatic amino acids have a hydrophilic part, which is able to interact with the heads of the lipids, as well as an apolar aromatic ring, which can interact with the hydrophobic tails of the lipids. Thus, anchoring and stabilizing the TMP within the membrane bilayer. Second, the so-called positive-inside rule [17, 18]. It refers to the prevalence of positively charges amino acids (ARG, LYS) in the cytoplasmic non-membrane regions of a TMP. First discovered in signal peptides [19], it was found to be true also for TMH. This charge-bias has a strong influence on the orientation of TMPs within the membrane and changing it through mutations can be enough to flip it [20, 21]. Computational prediction methods frequently make use of those features, especially the positive-inside rule.

1.1.6. Transmembrane Protein Topology

The topology of a TMP typically refers to how all residues of a protein sequence are positioned in relation to the membrane, i.e., which parts of the sequence cross the lipid bilayer, and which parts are situated outside the membrane and on what side of it. Another common term is the inside/outside topology, which refers to whether a particular part of the protein sequence is "inside" or "outside" relative to the membrane. Here, inside and outside most often refer to cytoplasmic and extracellular regions, respectively, though it depends on the type of membrane. For example, the membrane of a cell organelle has no extracellular side. In this case, outside would usually refer to the parts within the organelle.

1.2. Membrane Protein Structures

Due to the prevalence of transmembrane proteins (TMP) in nature and their importance to the pharmaceutical industry (many drug targets are TMPs) [1, 22], you would expect there to be plenty of well-resolved 3D structures. However, quite the opposite is the case. With less than five percent of all structures in the Protein Data Bank [23] (PDB) being TMPs, they are significantly underrepresented [24–26]. The simple reason is that experimentally determining the 3D structure of TMPs is harder than for most other proteins [27]. For one, overexpression of TMPs can be toxic to the cell and it can lead to them forming inclusion bodies [28]. In contrast to other proteins, which can fold on their own, TMPs often need a lipid bilayer for correct folding of their 3D structure. This makes the experimental determination more difficult, especially with classical X-ray crystallography. Some tried to stabilize the TMPs with the help of antibodies or fusing them with soluble domains [29–31]. However, this might alter their conformation. Recently, cryo-electron microscopy [32] gained popularity, which does not require crystallization. Unfortunately, it comes at the cost of having a lower resolution. Despite this rarity of 3D structures for TMPs, there are a few dedicated databases offering easy access and additional annotations not found in PDB. Two of the more frequently used databases are OPM [5] and PDBTM [33–35] (see below).

1.2.1. Orientations of Proteins in Membranes (OPM) database

As of early 2023, the Orientations of Proteins in Membranes [5] (OPM) database contains 8,073 PDB entries, representing 4,235 distinct protein structures. Only 664 of those PDB entries (distinct structures: 391) are beta-barrel TMPs, the others are alpha-helical TMPs. The OPM database is automatically updated in regular intervals using the PPM method [36–38]. For this, new PDB entries are downloaded from PDB and processed by PPM. The method tries to find the best positioning of the PDB structures in a membrane bilayer by calculating the transfer energy needed for inserting the protein into the membrane. PPM tests several models of membranes with varying thickness and features. The method itself received several improvements over the years, currently being in its third iteration [38]. OPM also offers a web service to upload your own 3D structure and apply the PPM method. OPM classifies its entries in a hierarchical system. The first level describes the type of membrane protein, i.e., whether it is transmembrane, monotopic or peripheral, or a membrane-active peptide. The second level provides information about the main secondary structure composition: all-alpha, all-beta, alpha+beta, alpha/beta, and non-regular proteins. The third level is the superfamily, i.e., proteins that are evolutionary related and have superimposable 3D structures. The fourth, and final, level is family, which includes proteins with detectable sequence homology. Each OPM entry lists the number and position of its transmembrane segments, though it does not explicitly distinguish between transmembrane beta-strands (TMB), transmembrane helices (TMH), or other membrane regions. However, this information can be inferred from the type of TMP and the local secondary structure. The PDB structures themselves are modified by PPM in such a way that the membrane bilayer is parallel to the z-axis, with its center at the origin of the coordinate system. Planes of nitrogen and oxygen atoms are added, representing the inner and outer membrane boundary, respectively.

1.2.2. Protein Data Bank of Transmembrane Proteins (PDBTM)

The Protein Data Bank of Transmembrane Proteins [33–35] (PDBTM) is an automatically updated repository of PDB structures containing TMPs, similar to OPM. At the time, it has 8,142 structures (7,611 alpha-helical TMPs, 526 beta-barrel TMPs). PDBTM uses the TMDET [39] method to embed the protein structures into a simulated membrane. It calculates the optimal positioning within the membrane based on the accessible surface area of all residues that would be exposed to the membrane. TMDET

is periodically run against new 3D structures in PDB to filter for structures of TMPs. Putative TMP structures are manually curated, i.e., visually inspected, after the automatic detection. Unlike the detailed classification of TMPs found in OPM, the entries in PDBTM are simply classified by their main structural elements passing the membrane: alpha, beta, and coil for undetermined elements. In contrast to OPM, PDBTM carefully annotates all regions of the protein sequence, not just the membrane-crossing segments. Those annotations include beta-strand, alpha-helix, coil, membrane-inside, membrane-loop, interfacial helix, as well as side 1 and side 2. Beta-strand, alpha-helix, and coil represent TMBs, TMHs, and other transmembrane regions, respectively. Membrane-inside are parts of beta-barrel TMPs that are inside the beta-barrel itself and do not touch the membrane, membrane-loops are re-entrant loops that dip into the membrane but do not fully cross it, and interfacial helices are alpha-helical regions close and mostly parallel to the membrane surface. Finally, sides 1 and 2 represent the other non-membrane parts of the protein sequence and, if possible, are mapped to the corresponding inside/outside topology. Unlike in OPM, the PDB structures are not modified to nicely fit the membrane. However, the PDBTM entries contain the necessary information to do so if desired.

1.3. Classic Computational Prediction Methods

In an attempt to alleviate the gap of known 3D structures for transmembrane proteins (TMP), countless prediction methods have been developed and published within the last three decades (Table 1.1). Though most of those methods only predict the position of membrane segments within a protein sequence, instead of the actual 3D structure, this information is often good enough for many research projects. Further features predicted by many methods include signal peptides, membrane re-entrant loops, and the inside/outside topology of the non-membrane regions, i.e., on which side of the membrane they are located. Inside is most often synonymous with cytoplasmic and outside with extracellular, though this can vary depending on the type of membrane the protein is located in. While the first prediction methods started quite simple from first principles, authors adapted more and more sophisticated and complex concepts and machine learning architectures over the past decades. In the next sections, I will present how those methods evolved from very simple programs to complex and powerful deep learning models. It is important to note that the following is not a complete list of all

membrane prediction methods, as there are just too many. Instead, I selected them based on a combination of innovation, popularity within the scientific community, and my personal experience.

1.3.1. Prediction from First Principles

The very first membrane prediction methods made use of the hydrophobic nature of the lipid bilayer and consequently the transmembrane segments of the protein sequences. In 1992, Gunnar van Heijne published TOP-PRED [40]. While being extremely simple, its algorithm is able to detect transmembrane helices (TMH) within protein sequences and predict their inside/outside topology. First, a window of 21 residues is shifted over the whole protein sequence to calculate the average hydrophobicity along the sequence, i.e., for all overlapping segments of 21 residues, a hydrophobicity score is reported. In order to achieve a slight smoothing effect, the first and last five residues of the window contribute less to the overall average than the central 11 residues, with the magnitude of all position-weights forming a trapezoidal shape. An empirically chosen threshold then distinguishes between certain and putative membrane segments, i.e., regions with a high average hydrophobicity score are categorized as certain segments, while regions with a moderate score are putative. Next, all possible topologies that include all certain segments and any number of the putative ones are generated and scored based on the positive-inside rule. This means that the number of positively charged amino acids (ARG, LYS) on each side of the putative membrane boundary are counted. The model with the highest charge bias towards one side is then selected, with inside being assigned to the side with more positive charges. TopPred II [45] slightly improved upon TOP-PRED by introducing additional rules aimed at the inside/outside topology prediction for eukaryotic proteins, though the actual detection of TMHs stayed the same. Another early method, SOSUI [51], similarly uses a running average hydrophobicity to detect TMHs. In addition, it considers the amphiphilicity of the end region of a putative TMH. This way it is able to better distinguish between true TMH and other helices with high average hydrophobicity or region in the protein core.

1.3.2. Machine Learning for Membrane Protein Prediction

Although the early prediction methods produced reasonably good results (for their time), their simplicity was heavily influenced by human bias. This includes the very narrow

Table 1.1.: Selection of several transmembrane protein (TMP) prediction methods of the last three decades. **Year** indicates when the method was first published. **Model** provides information about whether the method uses a simple hydrophobicity threshold, a statistical model with propensities for each amino acid, a machine learning model (CRF: conditional random field, GRHCRF: grammatical-restrained hidden conditional random field, HMM: hidden Markov model, LSTM: long short-term memory network, NN: neural network, RF: random forest, SVM: support vector machine), or a consensus prediction of multiple other methods. **Input** shows whether the method utilizes only the information contained in the protein sequence, some form of evolutionary information (e.g., MSA, profile, or PSSM), or a protein language model (pLM). **Type** specifies the structural class of TMPs predicted by the method, i.e., alpha-helical TMPs or beta-barrel TMPs. **SP** and **RL** indicate if the method also predicts signal peptides or re-entrant loops, respectively.

Name	Year	Model	Input	Type	SP	RL
TOP-PRED [40]	1992	Hydrophobicity	Sequence	Alpha	No	No
MEMSAT [41]	1994	Statistical	Sequence	Alpha	No	No
PHDhtm [42–44]	1994	NN	Evo. Inf.	Alpha	No	No
TopPred II [45]	1994	Hydrophobicity	Sequence	Alpha	No	No
MEMSAT2 [46]	1998	Statistical	Sequence	Alpha	Yes	No
TMHMM [47, 48]	1998	HMM	Sequence	Alpha	No	No
HMMTOP [49, 50]	1998	HMM	Sequence	Alpha	No	No
SOSUI [51]	1998	Hydrophobicity	Sequence	Alpha	No	No
Phobius [52]	2004	HMM	Sequence	Alpha	Yes	No
PROFtmb [13, 53]	2004	HMM	Evo. Inf.	Beta	No	No
PolyPhobius [54]	2005	HMM	Evo. Inf.	Alpha	Yes	No
MEMSAT3 [55]	2007	NN	Evo. Inf.	Alpha	Yes	No
OCTOPUS [56]	2008	NN + HMM	Evo. Inf.	Alpha	No	Yes
SPOCTOPUS [57]	2008	NN + HMM	Evo. Inf.	Alpha	Yes	Yes
MEMSAT-SVM [58]	2009	SVM	Evo. Inf.	Alpha	Yes	Yes
TOPCONS [59]	2009	Consensus	Evo. Inf.	Alpha	No	No
BOCTOPUS [60]	2012	SVM + HMM	Evo. Inf.	Beta	No	No
BetAware [61]	2013	NN + GRHCRF	Evo. Inf.	Beta	No	No
TOPCONS2 [62]	2015	Consensus	Evo. Inf.	Alpha	Yes	No
CCTOP [63, 64]	2015	Consensus	Evo. Inf.	Alpha	Yes	Yes
TMSEG [65]	2016	RF + NN	Evo. Inf.	Alpha	Yes	No
BOCTOPUS2 [66]	2016	SVM + HMM	Evo. Inf.	Beta	No	No
BetAware-Deep [67]	2021	LSTM + GRHCRF	Evo. Inf.	Beta	No	No
DeepTMHMM [68]	2022	LSTM + CRF	pLM	Alpha + Beta	Yes	No
DeepTMpred [69]	2022	CNN + CRF	pLM	Alpha	No	No
TMbed [70]	2022	CNN	pLM	Alpha + Beta	Yes	No

selection of features, i.e., average hydrophobicity, positive charges, and later amphiphilicity, as well as their respective thresholds. Modeling the complexity of membrane protein folding with only a few "if-else" statements is prone to error. Thus, the next generation of prediction methods ventured into the field of machine learning and adapted models that are more complex: Neural networks (NN), support vector machines (SVM), and hidden Markov models (HMM), to name a few. Those architectures are able to automatically extract, learn, and model complex and non-linear relations between the individual residues in a protein sequence. However, one still needs to define a suitable set of input features for those models to work properly. A very basic approach is one-hot encoding (Figure 1.3 A). This means that every residue in the sequence is represented by a one-hot vector, i.e., a vector that contains only zeros except for a single one, which indicates the corresponding type of amino acid. Typically, those vectors have a length of 20, representing the 20 standard amino acids. However, sometimes those vectors are extended to represent additional symbols, like ambiguous or unknown amino acids. A slightly more sophisticated way to encode the input is to use the corresponding rows from the BLOSUM62 [71] (or similar) matrix to represent each amino acid (Figure 1.3 B). This provides the model with additional information about the relation between the different amino acids, i.e., which amino acids have similar properties and are likely substitutes. Further, it is often helpful to explicitly provide the model with relevant features based on the task. For membrane prediction, this would include hydrophobicity, charge, and polarity of the amino acids.

Finally, it is usually beneficial to consider the local sequence context, instead of only a single residue, when making a prediction for a single residue. The most common approach for this is the sliding window method. For a predefined window size, all features are concatenated into a single input vector and processed by the model to make a prediction for the central residue of the window. For example, a sliding window of size 15 with its features based on the BLOSUM62 matrix would contain a total of $15 \times 20 = 300$ features, i.e., each of the 15 residues within the window being represented by its 20 BLOSUM62 scores. The prediction of the central residue, i.e., the seventh residue, would then be based on all 300 features. By sliding such a window over the whole sequence, all residues within a protein sequence can be processed. However, predicting the first and last residues within a sequence requires extra care as the window would hang over either the start or end of the protein sequence. This is often addressed by adding an additional feature value to represent padding positions outside the actual

amino acid sequence, or by setting all feature values to zero for those positions. In essence, the sliding window approach mimics a 1D convolutional kernel.

A

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
E	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

B

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
S	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-2
E	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
Q	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1
W	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	2
E	-1	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 1.3.: Two different encodings of the amino acid sequence `SEQWENCE`. **A:** One-hot encoding represents each residue in the sequence with a vector of length 20. The amino acid present at each sequence position is indicated by a 1 in the corresponding vector element, all other elements are 0. **B:** Residues in the sequence represented by the corresponding row from the BLOSUM62 [71] matrix. In both cases, only the encoding of the first five residues is shown. Note that the same type of amino acid is always represented by the same vector (e.g., both `E` in the depicted encodings).

1.3.3. Leveraging Evolutionary Information

Arguably, one of the most impactful improvements to encode protein sequences was leveraging so-called evolutionary information or profiles. While one-hot encoding or BLOSUM62-like substitution matrices always provide the same fixed representation for

a particular type of amino acid, evolutionary profiles hold specific information about the conservation or variability of each individual residue within a protein sequence (Figure 1.4). The main idea behind this approach is that proteins from the same protein family have similar sequences, sharing structural and functional features. Especially regions in the sequence that are crucial for the structure and function of the protein are usually highly conserved between members of the same protein family. On the other hand, regions with a high variability of amino acids are most likely less relevant for the overall function of the protein. In the context of membrane proteins, this means that the hydrophobic characteristics of transmembrane regions are usually more conserved than the more variable short non-membrane loops in-between them. This provides the machine learning models with valuable evolutionary information that would be missing in the previous single-sequence encodings.

A common method to generate such evolutionary profiles is to search the query sequence of interest against a big database of protein sequences and extract those that are sequence-similar. This is usually accomplished by using optimized search programs such as PSI-BLAST [72], HHblits [73], or MMseqs2 [74–76]. Once similar sequences are collected they are aligned with the query sequence to compute a multiple sequence alignment (MSA). The MSA is then further processed to compress its content into sequence profiles or position-specific scoring matrices (PSSM [72, 77]). A sequence profile typically contains the raw distribution of amino acids per position of the query sequence, i.e., the relative frequency indicating how often a particular amino acid appeared at a specific position within the MSA (Figure 1.4 B). For a PSSM, the frequencies in the sequence profile are normalized according to a background distribution (e.g., the background frequencies used in the BLOSUM62 matrix), and converted into log likelihoods (Figure 1.4 C). Thus, a positive score in a PSSM indicates an amino acid that is likely to appear at this specific position, while a negative score indicates a rather undesired amino acid substitution. Those scores are well suited to be understood and processed by a machine learning model and provide valuable information, not only about the single input sequence but the whole protein family. However, the downside of using MSAs, sequence profiles, or PSSMs as input is the computational overhead associated with it. Searching against a sufficiently large database with millions or billions of protein sequences requires either vast amounts of computational resources or time, as does building large MSAs from hundreds or thousands of sequences. Further, each of the steps involved in the database search and MSA construction requires careful tuning of the parameters used by the employed programs. Finally, this approach might simply fail for protein

sequences without known homologs. Nevertheless, many of the most successful methods for predicting TMPs utilize evolutionary information in one form or another.

1.3.4. Membrane Prediction Methods of the Last 30 Years

The MEMSAT Family

The original MEMSAT [41] method is a statistical model to predict TMHs. Based on a data set of globular and membrane protein, the authors calculated the propensities (log likelihoods) for each of the 20 standard amino acids to be part of one of five structural states: inside loop, outside loop, inside helix end, outside helix end, and helix middle. Inside and outside loops refer to non-membrane regions of the protein sequence. Inside end, outside end, and middle of a helix refer to the caps and central residues of a TMH. The authors defined the helix caps as the first and last four residues at each end of a TMH. Separate statistics were computed for single-pass TMPs and multi-pass TMPs. Using those propensity scores, MEMSAT can calculate how well a specific topology fits to an amino acid sequence. As brute-forcing all possible topology combinations for a given sequence would be computationally too expensive, MEMSAT utilizes a dynamic programming algorithm similar to the Needleman-Wunsch algorithm used for pairwise alignments. MEMSAT2 [46] extends the previous method by adding statistics for signal peptides, reducing the number of false positive predictions. The next iteration, MEMSAT3 [55], adapts a neural network architecture, moving away from the purely propensity-based model, and uses PSSMs instead of single sequences as input. It also uses a sliding window of 19 residues, considering the sequence-adjacent residues when making a prediction. Finally, MEMSAT-SVM [58] changed the model architecture again, this time to support vector machines (SVM). The authors trained four binary SVMs, each specialized on a different type of structural state: TMH, inside and outside loop, re-entrant helix, signal peptide. While the three binary model for TMHs, re-entrant helices, and signal peptides predict whether a residues belongs to this class or not, the model for loops categorizes a residue as either belonging to an inside loop or outside loop region. A dynamic programming algorithm, similar to the one in previous versions of MEMSAT, is used to process the output of all four SVMs and predict the final topology for a protein sequence.

A

```
SEQWENCE
SEEWDNCE
AEQWDNCE
SEQW-NCE
```

B

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
S	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0
E	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	25	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
E	0	0	67	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

C

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
S	2	-1	-1	0	-2	0	-1	-2	0	-2	-1	0	-1	0	-1	4	1	-1	-3	-2
E	-1	-4	1	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-3	-3	-2
Q	-1	-3	0	3	-3	-2	0	-3	1	-2	-1	0	-2	5	1	0	-1	-2	-3	-2
W	-3	-4	-4	-3	1	-3	-2	-2	-3	-2	-2	-3	-5	-2	-3	-3	-2	-3	11	2
E	-1	-2	5	3	-2	-1	-1	-2	0	-2	-2	1	-1	0	-1	0	-1	-2	-2	-2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 1.4.: Two different encodings of a multiple sequence alignment (MSA) for the query sequence `SEQWENCE`. **A:** MSA with the query sequence and three similar sequences. **B:** Query sequence encoded as a sequence profile containing the relative frequencies (percentages) of amino acids present in each MSA column. **C:** Position-specific scoring matrix (PSSM [72, 77]) for the query sequence based on the MSA and the BLOSUM62 [71] background frequencies. Each value represents the log likelihood of the corresponding amino acid substitutions. In both cases, only the encoding of the first five residues is shown.

PHDhtm

PHDhtm [42] is one of the first prediction methods to combine machine learning and evolutionary information in order to predict TMHs and the overall topology of membrane proteins. It consists of a multi-level prediction system, including two neural networks (NN) and an empirical filter. The input of the first NN is a sliding window of the MSA for the protein sequence, as well as a few global statistics. The MSA statistics are the relative amino acid frequencies for each of the MSA columns included in the sliding window, the frequencies for insertions and deletions, and a conservation score for each position. Thus, 24 features encode each position: 20 standard amino acids, 1 indicator for padding, and 3 for insertions, deletions, and conservation. The global statistics include the overall amino acid composition for the protein sequence, its total length, and the distances from the N- and C-terminus to the current sliding window. The input to the second NN is comprised of sliding window over the output of the first NN, plus the conservation scores, and the same global statistics used for the first NN. Four features encode each position in the sliding window: 2 output values from the first NN indicating TMH/non-TMH, 1 for padding, and 1 for conservation. Finally, an empirical filter is applied to the output of the second NN (TMH, not-TMH), deleting or extending TMHs with less than 17 consecutive residues, and splitting very long TMHs with more than 35 residues. The inside/outside topology is determined by the difference of positive charges between the two sides separated by the predicted TMHs. The main advantage of the two-level NN model is that the first NN can learn to detect residue belonging to TMHs (i.e., sequence-to-structure), while the second NN learns to model TMHs as segments of consecutive residues and learns their approximate lengths (i.e., structure-to-structure). In a later version of PHDhtm [43, 44], the authors replaced the empirical filter with a dynamic programming algorithm. First, all possible combinations of TMHs with 18-25 residues are generated, and each TMH is scored according to the average TMH-propensity of its residues as predicted by the second NN. Next, the highest-scoring non-overlapping TMHs are selected one-by-one and iteratively added to the predicted topology model. Each intermediate topology model is scored according to how well the assigned residue-states fit the predicted propensities of the second NN. The topology model with the highest score is selected for the final prediction, though lower-scoring models are shown, too.

TMHMM & HMMTOP

TMHMM [47, 48] is one of the first membrane prediction models to implement a hidden Markov model (HMM). First described in 1998 and later republished in 2001, it is still frequently used today. Unlike NNs or SVMs, which independently predict the states of individual residues, HMMs try to estimate the best overall state-predictions for the whole sequence based on a pre-defined grammar. This fit well with the segmented structure of TMPs, where the non-membrane regions of the protein sequence are separated by TMHs crossing the membrane. Instead of having to apply additional filters to the predictions in order to prevent too long or too short TMHs, the model itself can enforce those parameters. The authors chose a cyclic HMM, forcing the topology to change between inside and outside after each TMH. The HMM used in TMHMM contains structure-blocks to model the core regions of TMHs, the cytoplasmic and non-cytoplasmic caps of those TMHs, cytoplasmic loops, non-cytoplasmic short and long loops, and globular domains between loops. The helix core model consists of 25 sequential states, including shortcuts, which enforces lengths of 5-25 residues. The helix caps each have five states, resulting in TMHs with lengths of 15-35 residues. Each loop region is modeled by 20 interconnected states, and together with the single self-looping globular domain state, this allows for non-membrane regions of arbitrary lengths. HMMTOP [49] was developed independently around the same time as TMHMM. It implements a similar HMM architecture, though the exact parameters (e.g., minimum and maximum lengths) differ slightly. HMMTOP [50] was later improved by allowing users to pre-define the states for parts of the protein sequence. This often resulted in better predictions and prevented incorrect predictions for already well-annotated parts of a protein sequence.

Phobius & PolyPhobius

Phobius [52] is another early HMM-based prediction method for membrane proteins. It was the first one to model signal peptides and TMHs in a single HMM. Essentially, it is a combination of the HMMs implemented in TMHMM and SignalP-HMM [78], though with slight modifications to the exact number of states. In addition to the previously described states in TMHMM, Phobius includes separate states to model the n-, h-, and c-regions of signal peptides, as well as its cleavage site. This enables Phobius to distinguish between TMHs and the hydrophobic h-regions found in signal peptides, which also often form helical structures. With this combined HMM, Phobius reaches better prediction

performances than TMHMM, HMMTOP, and SignalP-HMM alone, or combinations of those methods. A few years later, the same authors published an improved method, which they called PolyPhobius [54]. Its main advantage over Phobius is the ability to process aligned sequences from a MSA, instead of only a single sequence. PolyPhobius first predicts the label probabilities for each individual sequence in the MSA, and then averages the probabilities for aligned residues to get the predictions for the original query sequence. Leveraging this evolutionary information, PolyPhobius outperforms the already quite well performing Phobius. Although almost two decades old, both methods are still frequently used today.

OCTOPUS & SPOCTOPUS

OCTOPUS [56] had one of the most complex model architectures at the time. Combining elements from previous successful methods, OCTOPUS is made up of five different NNs plus a HMM. As input, it uses both a PSSM and a raw amino acid frequency profile generated by a PSI-BLAST [72] search. The authors trained four separate NNs to predict the preference of a residue to be in one of four structural states: membrane, interface, loop, and globular; each NN specialized for one of those states. The states are defined based on the distance of a residue from the membrane center: less than 13 Å (membrane), 11-18 Å (interface), 13-23 Å (loop), and more than 23 Å (globular). Thus, a single residue can belong to one or two states. The input for those NNs are sliding windows over the PSSM. The fifth NN was trained to predict the inside/outside topology preference of a residue based on sliding windows over the average frequency profiles for the amino acids ARG+LYS, and TYR+TRP, i.e. focusing on the positive-inside rule and aromatic belt. The output values from all five NNs are then combined and used as emission probabilities for the HMM. The HMM itself distinguishes between 10 different states: inside and outside states for each of transmembrane, hairpin, loop, globular, and re-entrant/dip. Transmembrane, hairpin, and re-entrant/dip states differ by how far they are reaching into the membrane bilayer. Re-entrant/dip regions reach only partially into the membrane, hairpins come close to the opposite border of the membrane bilayer, and transmembrane regions completely emerge on the other side. This made OCTOPUS one of the few methods to predict membrane regions other than TMHs. In a later update, the authors added an additional NN and HMM to predict signal peptides. This new method is called SPOCTOPUS [57].

TMSEG

TMSEG [65] is a multi-stage prediction method, combining two random forests (RF), a NN, and an empirical filter. The first stage is a RF that takes as input a sliding window of 19 residues over a PSSM generated by PSI-BLAST [72], local hydrophobicity, charge, and polarity, as well as global statistics similar to PHDhtm (overall amino acid distribution, sequence length, distance to N- and C-terminus). It predicts the preference for a residue to belong to one of three states: TMH, signal peptide, or other. In the next stage, a median filter smooths the output of the RF, and an empirical filter removes TMHs and signal peptides that are too short. The third stage consists of a NN trained to distinguish between correct and incorrect TMHs. It is used to optimize the exact placement of the already predicted TMHs, and to split very long TMHs into two separate TMHs. In the final stage, another RF determines the inside/outside topology of the protein sequence. For this, amino acid distributions and percentage of positively charged residues are calculated for each of the two sides determined by the predicted TMHs, as well as the absolute difference in positive charges. The RF then predicts the N-terminal inside/outside topology, which in turn is used to infer the topology for all other sequence regions in-between the predicted TMHs. Besides its good prediction performance, the modularity of TMSEG allowed stages 3-4 to be applied to the prediction results of other methods, often improving those as well. TMSEG is described in more detail in Chapter 2.

Consensus Methods: TOPCONS & CCTOP

Another approach to developing and training a completely new machine learning model is the construction of consensus methods based on previous membrane prediction methods. Such consensus methods are often more accurate and stable in their prediction output than any of the individual methods alone. One example is TOPCONS [59], which combines the prediction output of SCAMPI-single [79], SCAMPI-multi [79], PRO-TMHHMM [80], PRODIV-TMHMM [80], and OCTOPUS. After generating the results for each method, they are used to calculate a topology profile for the sequence, containing the average preferences for each residue to be either part of a TMH, inside (cytoplasmic), or outside (extracellular). This profile is then processed by a three-state HMM to find the overall best topology prediction. TOPCONS2 [62] is an improved version that replaces the previous prediction methods with Philius [81], SCAMPI-multi, OCTOPUS,

SPOCTOPUS, and PolyPhobius. As some of those methods are able to predict signal peptides, the HMM was adjusted accordingly and includes four states instead of three. Another example for a consensus method would be CCTOP [63, 64], combining a total of ten methods: HMMTOP, MemBrain [82], MEMSAT-SVM, OCTOPUS, Philius, Phobius, PRO-TMHMM, PRODIV-TMHMM, SCAMPI-multi, and TMHMM.

PROFtmb

PROFtmb [13, 53] differs from the previous membrane prediction methods by focusing on the less common beta-barrel TMPs and their transmembrane beta strands (TMB). The method implements a HMM with a total of 91 states, modeling TMBs, periplasmic loops, and extracellular loops. Although the HMM architecture is quite straightforward, the authors note a few key features that increased the prediction performance of PROFtmb. First, they model TMBs crossing the membrane from the periplasmic side to the extracellular side and those crossing the other way around separately, instead of having them both share parameters. Residues within a TMB always alternate between a lipid-facing and pore-facing orientation. Next, due to their prevalence in the data set, the HMM explicitly models short beta-turns with four or five residues on the periplasmic side, in addition to longer loops. Finally, the HMM is able to process sequence profiles instead of single sequences, improving the prediction performance of PROFtmb.

BetAware & BetAware-Deep

BetAware [61] predicts beta-barrel TMPs and their TMBs in two steps. First, a NN architecture applied to the protein sequence predicts whether it is a beta-barrel TMP or not. If it is, a grammatical-restrained hidden conditional random field [83] (GRHCRF) predicts the three-state topology of the protein, i.e., inside, outside, and TMB regions. The advantage of using a GRHCRF over a normal conditional random field (CRF) is that the output of the former can be guided by a pre-defined grammar, similar to a HMM, thus preventing it from generating biologically meaningless predictions. Like many of the previous methods, BetAware uses sequence profiles as input to increase its prediction performance. Just recently, an updated version called BetAware-Deep [67] has been released. The main improvements are the inclusion of a hydrophobic moment along the sequence, the use of a bi-direction long short-term memory (LSTM) network, and

extending to five states. The hydrophobic moment measures the change in hydrophobicity along the protein sequence and is based on the amino acid frequencies in the input sequence profile. The LSTM takes the sequence profile and hydrophobic moment as input and predicts the per-residue probabilities for each of the five states: non-barrel region, periplasmic, extracellular, transmembrane, and extended beta strand. The non-barrel region represents the parts of the protein that come before or after the beta-barrel structure. The extended beta strands are the parts of the TMBs which extend out of the membrane, i.e., they are no longer classified as transmembrane but still part of the contiguous beta strand. The GRHCRF then takes the sequence profile and predicted state-probabilities as input and predicts the overall protein topology. Finally, the five states are converted to the typical three states by mapping the non-barrel regions to periplasmic regions and the extended beta strands to TMBs.

BOCTOPUS & BOCTOPUS2

BOCTOPUS [60] takes the successful architecture of its predecessors (OCTOPUS, SPOCTOPUS) and applies them to beta-barrel TMPs. However, BOCTOPUS uses SVMs instead of NNs. First, three separate SVMs take a PSSM as input and predict the per-residue preferences for either inside, outside, or TMB. Those prediction values are then fed into a HMM to generate the final topology prediction. BOCTOPUS2 [66] improves upon its predecessor by splitting the TMB SVM into two separate SVMs, one for lipid-facing residues and one for pore-facing residues. An empirical filter then processes the predictions of the now four SVMs to remove likely false positives, i.e., predicted TMBs far away from all other TMBs. Finally, the HMM uses the filtered SVM outputs and predicts the four-state topology, i.e., inside, outside, lipid-facing, and pore-facing.

1.4. Rise of the Transformer and Protein Language Models

When Vaswani *et al.* published their famous paper with the title *Attention Is All You Need* in 2017 [84], it had a far-reaching impact on the field of Natural Language Processing (NLP) and Machine Learning in general. In the publication, they presented their new Transformer model architecture and showed its superior performance on several machine translation tasks. However, as the Transformer has no strong inductive biases, i.e., it does not make strong assumptions about the structure of the input data, it was

soon adapted to other domains such as images [85], audio [86], and most importantly, proteins [87–93].

1.4.1. The Transformer

The Attention Mechanism

The main component responsible for the good performance of the Transformer is the so-called attention mechanism. It enables the Transformer to consider all previous and future elements of a given input sequence when performing computations, i.e., achieving true bi-directionality. In general, attention works by comparing and aggregating features for all tokens in a sequence (e.g., words in a sentence). Given a sequence of token vectors x_i, the overall sequence can be represented as a matrix X of size $L \times D$, where L is the length of the sequence and D is the size of each token vector. In the first step of the attention mechanism, three matrices Q, K, and V are generated from the input matrix X. They are also often referred to as query (Q), key (K), and value (V). The Q, K, and V matrices are computed by multiplying X with the weight matrices W_q, W_k, and W_v, respectively, which are learned during training. Each weight matrix is of size $D \times d_k$, thus projecting X to query, key, and value matrices of size $L \times d_k$. Next, the Q matrix is multiplied with the transposed K matrix, resulting in a matrix S of size $L \times L$, which represents the pairwise similarity between all row-vectors of Q and K. To prevent the individual dot-products in matrix S from having too large magnitudes, the matrix is scaled by the inverse of $\sqrt{d_k}$, hence this is also called scaled dot-product attention. The matrix S is then processed by applying a row-wise softmax operation, thereby normalizing each row to a total sum of 1. Finally, the normalized matrix S is multiplied with V to generate the output of the attention mechanism. Mathematically, this can be written as:

$$Attention(Q, K, V) = Softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \tag{1.1}$$

This operation is often compared to the process of comparing a set of queries (Q) against the keys (K) of a database and retrieving the corresponding values (V). The weight matrices W_q, W_k, and W_v extract the necessary features from the input X to represent the relationship of the input tokens. In the output of the attention mechanism, each

token is represented by a weighted sum of the values (V) of all tokens in the sequence. The weights are calculated by comparing the query (Q) and key (K) representations of the tokens.

This attention mechanism is usually extended to a so-called multi-head attention. Instead of calculating a single Q, K, and V matrix for the input X, multiple of each are computed. Every set of corresponding Q, K, and V matrices is referred to as an attention head, and each head is using its own W_q, W_k, and W_v matrices. After calculating the attention output for every head, the resulting matrices are concatenated to a single matrix. This matrix is then multiplied with a learned weight matrix W_o to combine the individual attention head results and compute the output of the multi-head attention mechanism. This extension of the attention mechanism allows each attention layer to consider multiple relationships between all input tokens at once, instead of focusing on only one at a time.

Further subcategories of attention mechanisms are the so-called self-attention and cross-attention. In the former case, all Q, K, and V matrices are computed from the same input X (as previously described). However, in the case of cross-attention, the query (Q) matrix is computed from a different input than the key (K) and value (V) matrices. For example, in the case of machine translation (e.g., English to German), Q is computed from the target sentence (e.g., "Der Himmel is blau"), while K and V are computed from the source sentence (e.g., "The sky is blue"). This allows each of the tokens in the target sentence to attend to the corresponding tokens in the source sentence.

Positional Encodings

One critical and often undesired aspect of the attention mechanism is that it is equivariant regarding the order of the tokens. This means that the attention mechanism alone would not be able to distinguish between the two sentences "the mouse ate the cheese" and "the cheese ate the mouse", or any other permutation of those five words. In the original Transformer publication [84], the authors proposed encoding the absolute position of a token within the sequence by using a combination of sine and cosine functions. Those sinusoidal encodings are then added to the original input token vectors, thereby allowing the Transformer to learn and model relationships of tokens based on their absolute and relative positions. Later Transformer variants implemented different ways of encoding positional information. Some opted to add either fixed [94] or learned [95]

biases based on the relative positions of the tokens to the similarity matrix, just before the softmax function, which allowed the Transformer to take the relative order of tokens into account. Though those did not encode the absolute position of a token within the sequence, they showed very good results in practice. Rotary positional encodings [96] on the other hand applied rotations to the key and value vectors based on the absolute position of a token. Similar to the sinusoidal encodings, rotary encodings allow modeling absolute and relative relations between tokens.

Attention Is Not All You Need

The other main component of the Transformer is a standard two-layer feedforward neural network (FNN) that processes each token vector individually and refines their features. Those two components, an attention layer followed by a FNN, form what is called a Transformer block. Finally, a typical Transformer is made up of multiple such blocks. Thus, the model alternates between aggregating features across all tokens in the attention layer and then processing the new features of each token individually in the FNN.

Encoders and Decoders

Transformers can be further classified into encoder and decoder models, with the main difference being in how the attention mechanism is configured. An encoder Transformer utilizes standard self-attention in its attention layers, thus processing the input sequence bi-directionally. This configuration is often used when the main purpose of the model is to encode an input sequence into an information-rich and contextualized vector-representation. The output of an encoder is then used as the input feature for another downstream model, e.g., for classification tasks. On the other hand, a decoder Transformer applies a so-called causal attention mask to the self-attention layers, which sets the upper triangle (excluding the diagonal) of the similarity matrix S to negative infinity. This effectively prevents any individual token from attending to any of the tokens that are further down the sequence, i.e. the first token can only attend to itself, the second token only to itself and the first token, and so on. Thus, causal attention replicates the way a unidirectional long short-term memory (LSTM) network works. Decoders are typically used for generative tasks, e.g., generating new text by iteratively predicting the next word in a sentence based on all previous words. Finally, encoder and decoder models can be combined via cross-attention. Here the decoder has additional

cross-attention layers, which combine the token vectors of the decoder (queries) with the final output of the encoder (keys, values). Such encoder-decoder models are often employed for tasks that mix inputs from different domains. For example, when translating from one language into another, or when going from image to text (image captioning). In fact, the original Transformer published in 2017 [84] was an encoder-decoder model that used the encoder to process the source sentence of a machine translation task and the decoder to generate the translated target sentence.

Training a Language Model

Transformer-based language models (LM) can be trained using supervised methods, just like most other machine learning models. For example, an encoder-decoder LM intended for machine translation is trained on pairs of sentences from the source and target languages. However, there is a major benefit in self-supervised pre-training of LMs. This means that the LM trains on unlabeled data, which is usually much more available (e.g., by crawling Wikipedia) than labeled data for a specific task. The purpose of this pre-training is for the LM to learn the basic grammar and semantics of the language. Afterwards, the LM can be further fine-tuned on labeled data for its final task, if necessary. Two methods for self-supervised pre-training are Masked Language Modeling (MLM) and Autoregressive Language Modeling (ALM).

The idea behind MLM is similar to a cloze test, i.e. filling blanks in a sentence. The BERT [97] (Bidirectional Encoder Representations from Transformers) model was one of the first Transformers trained using MLM. Here, 15% of the input tokens were randomly masked, i.e. replaced by a special mask token, and the encoder-only model was tasked with reconstructing the masked tokens based on the remaining unmasked tokens. Further, to prevent the model from focusing solely on reconstructing the masked tokens, 10% of the masked tokens were instead replaced another random token from the vocabulary and another 10% were reverted to the original token. This forced the BERT model to learn to recognize correct or misplaced words and either pass them through to the output (for original tokens) or replace them with a correct word (for random replacements). The hidden representations, often called embeddings, just before the final output projection (back to token labels) can then be used for downstream tasks. For example, classification of either individual tokens or the overall sequence.

On the other hand, ALM is based on predicting the next token in a sequence. Popular models trained on ALM include the GPT [98–100] family by OpenAI. During ALM training, a special start token is prepended to the input sequence and the model is tasked to output the input sequence, but shifted one position to the left. This means that the first original token of the input is aligned with the special start token, the second original token with the first, and so on. Further, causal attention masking is applied to the self-attention layers of the model, which are often decoder-only models. Thus, each output token can only attend to the tokens that came before it in the sequence, essentially modeling a next token prediction task. During inference, the decoder would then start with a sequence, which might only consist of the special start token, and output a sequence that includes the next, most probable word. Then it would iteratively continue to use its own output as input, predicting the next tokens in the sequence, until it finally predicts the end of the sequence. Such decoder models are often used for generative tasks, such as auto-completion or text generation.

Finally, MLM and ALM training can also be combined in an encoder-decoder model. For example, the encoder gets a masked sequence as input and the decoder is tasked with predicting the reconstructed sequence in an autoregressive manner.

1.4.2. Protein Language Models

Adapting LM architectures from NLP to protein sequences can be quite straightforward. The simplest way is to consider every amino acid in the protein sequence as a single word (or token) and the whole sequence is equivalent to a sentence. Thus, the same training techniques can be applied to those so-called protein language models (pLM). Earlier pLMs showed promise when used to encode protein sequences to generate input features (embeddings) for downstream tasks [101]. However, the powerful Transformer architecture finally gave the needed push to put the quality of the embeddings generated by those pLMs on comparable levels to evolutionary information (e.g., PSSMs or MSAs).

The abundance of publicly available sequence data through databases also provides enough (unlabeled) training data for the self-supervised tasks of MLM and ALM. UniProtKB [102] gives easy access to over 200 million protein sequences, while lager databases, such as BFD [75, 103], hold up to 2.5 billion protein sequences. However, due to the nature of protein sequences, special attention should be paid to homology. Many of those protein sequences are highly redundant, which might bias the pre-training of the

model. This issue was tackled in several ways, to reduce the redundancy in the training samples. For example, Meta AI trained their ESM [91–93] family of pLMs on the less redundant UniRef50, UniRef90, and UniRef100 clusters, or combinations thereof. The ESM-1b model was trained on UniRef50, and ESM-1v on UniRef90. For the more recent ESM-2 models sequences were sampled from UniRef90 based on the UniRef50 clusters, in order to have a higher diversity of sequences without the full redundancy of UniRef90. Another pLM, ProtT5-XL-U50 [89], was instead first trained on the redundant BFD database and then fine-tuned on the less redundant UniRef50 cluster representatives.

Pre-training pLMs on large protein datasets enables them to learn the "grammar" and "semantic" of protein sequences. To successfully solve their task, which is often MLM, the pLMs must learn which amino acid motifs work in nature and which do not, or have not been observed yet. However, due to the global perception of the attention mechanism in Transformer models, this is not limited to local motifs, but can also extend to short- and medium-range interactions, e.g., between residues in alpha helices or beta strands. Even long-range interactions can be learned, which can happen between amino acids from different domains that are close in 3D space but distant in sequence space. Further, by training on multiple members of a protein family, the pLMs can recognize which regions are rather conserved and which are more variable.

Although the training of large pLMs can take several days or weeks, the big advantage lies in the fact that this has to be done only once. A pre-trained pLM encodes the compressed information from millions or billions of protein sequences in its trained weights. While the recent pLMs can be quite big (several GBs), modern hardware and optimizations for machine learning enable very fast computations. A Transformer-based pLMs can usually process several protein sequences per seconds. This often provides a speed-advantage over generating evolutionary information, which involves searching a database for similar sequences, aligning them into a MSA, and computing the desired statistics based on the MSA. Further, pLMs can also handle sequences for which there are few or no known homologs.

1.4.3. Protein Language Models for Membrane Prediction Methods

After the recent boom and success of modern pLMs in other fields of computational biology, such as the prediction of secondary structure [89] or function [104–106], pLMs have been tested for membrane prediction methods. Some of the more recent ones

are DeepTMHMM [68], DeepTMpred [69], and TMbed [70]. Taking advantage of the information-rich embeddings generated by pLMs, those methods reach state-of-the-art prediction performance without the need for MSAs or PSSMs, and instead work with single sequences as their input. Without having to search the query sequence against a huge database to generate sequence profiles, those pLM-based methods are able to process multiple sequences per second. For example, TMbed (see below) is able to predict the over 500,000 protein sequences in UniProtKB/Swiss-Prot in less than nine hours.

DeepTMHMM

DeepTMHMM [68] is the successor of TMHMM, one of the most popular prediction methods of the last two decades. In addition to improved prediction performance and the new capability to predict signal peptides, DeepTMHMM now also predicts beta-barrel TMPs, instead of only focusing on alpha-helical TMPs. It employs the ESM-1b [91] pLM to encode its input sequence and uses the generated embeddings as its sole input features. The architecture of DeepTMHMM is rather lightweight and consists of a bi-directional LSTM to process the embeddings and a conditional random field (CRF) to model the predicted topology. Similar to a hidden Markov model (HMM), a CRF can encode a certain grammar, which the predicted topology should adhere to. The authors decided on two main schemas, one for alpha-helical TMPs and one for beta-barrel TMPs. The alpha-helical model can start with either a signal peptide, an extracellular region, or a cytoplasmic region. This is followed by any number of transmembrane helices (TMH), each time changing from extracellular to cytoplasmic or vice versa. In contrast, the model for beta-barrels is more strict and forced to start with a signal peptide followed by a periplasmic region. The C-terminus is also restricted to the periplasm, enforcing an even number of transmembrane beta strands (TMB). Globular proteins are predicted by the alpha-helical model by simply not including any TMHs.

DeepTMpred

DeepTMpred [69] uses the ESM-1b pLM to generate its input features for a protein sequence, just like DeepTMHMM. However, DeepTMpred focuses solely on alpha-helical TMPs and does not predict beta-barrel TMPs or signal peptides. Its architecture consists of two sub-modules: one for TMH prediction and one for the inside/outside topology

prediction. The TMH sub-module uses a small convolutional neural network (CNN) and a CRF. While the single convolutional layer with a kernel size of three allows for only a small receptive field, the ESM-1b embeddings should already encode the needed global context. The CRF has only two states: TMH and non-membrane. The topology sub-module uses a lightweight attention mechanism. Unlike the attention mechanism in the Transformer, the attention layer chosen by the authors does not compare the individual residues with each other. Instead, the vector representations of each residue are projected to single weights and the residue representations are summed up according to their weights. This weighted sum of residue representations is then used to predict the N-terminal inside/outside topology. Further, only the first and last five residues of a protein sequence are used for the topology prediction, which according to the authors performed the best. Once the inside/outside topology for the N-terminus is determined, the topology of all sequence regions is extrapolated accordingly, alternating between inside and outside after each predicted TMH. The authors of DeepTMpred tested a modification of their model that included sequence profiles generated with PSI-BLAST [72] and HHblits [73], but they found that this sometimes even decreased the prediction performance. They speculate that it just adds noise to the overall input signal and that the ESM-1b embeddings already include the necessary information. Thus, the additional parameters make the model more complicated and less stable to train.

TMbed

TMbed [70] predicts alpha-helical and beta-barrel TMPs, as well as signal peptides. Unlike the previous two methods, it uses the ProtT5-XL-U50 [89] pLM to generate the input embeddings. The model architecture of TMbed consists of a small convolutional neural network (CNN), a Gaussian smoothing filter, and a Viterbi decoder. The CNN has four layers: a pointwise convolution to reduce the dimensionality of the input embeddings, followed by two parallel depthwise convolutional layers, and a final pointwise convolution. The CNN processes the input embeddings and predicts five state-probabilities for each residue: inside, outside, signal peptide, TMH, and TMB. A Gaussian filter then removes unwanted spikes in the prediction of the CNN and the Viterbi decoder models the final state-prediction for each residue in the protein sequence. In contrast to the CRF in DeepTMHMM, the Viterbi decoder in TMbed has only a few rulesets: 1) signal peptides must be at the start of a protein sequence, 2) signal peptides, TMHs, and TMBs must be at least five residues long, 3) the inside/outside topology must change

after each TMH and TMB. This allows TMbed to model more diverse protein topologies than DeepTMHMM. For example, it allows for beta-barrel TMPs with an uneven number of TMBs, or without a signal peptide at the start. It would even be able to predict proteins that have a combination of TMH and TMB segments, though thus far I am not aware of experimental evidence for such proteins. TMbed is described in more detail in Chapter 4.

1.5. 3D Structure Prediction Methods

Though not specifically designed for membrane proteins, general 3D structure prediction methods are vital tools for the *in silico* research of membrane proteins. By providing mostly reliable 3D structures, they can directly bridge the huge structure gap present in experimental 3D structure databases. However, they often do not come with labels attached, i.e. they usually do not predict whether a protein is a membrane protein or not. Nevertheless, the various sequence-based TMP prediction methods presented in the previous sections are perfectly suited as pre-filters to select only probable TMPs for the 3D structure generation. This is especially true for the very fast and accurate pLM-based prediction methods, such as DeepTMHMM [68] and TMbed [70]. They can filter whole databases with several millions of protein sequences within a matter of days, greatly reducing the downstream computational resources needed to perform the actual 3D structure prediction.

1.5.1. AlphaFold2

Without a doubt, AlphaFold2 [87, 88] made one of the biggest impacts in the field of computational biology in the last few years. Providing quite accurate and reliable 3D structure predictions for proteins, it single-handedly started to close the gap between 3D structure databases and protein sequence databases. As of 2023, AlphaFold DB [107] provides access to hundreds of millions of pre-computed 3D protein structures. Though an accomplishment of its own, AlphaFold2 is also another offshoot of the recent Transformer [84] revolution. Several of its key components make heavy use of the attention mechanism to aggregate and process the information contained within the input MSA and its individual protein sequences, as well as to model the interactions between the

atoms of the predicted structure. Combining this powerful new machine learning architecture with the information-rich input of evolutionary information (MSAs), gave rise to this milestone of 3D structure prediction. While not designed to model a membrane bilayer, research showed that AlphaFold2 is capable of reliable prediction of membrane protein structures [108].

1.5.2. ESMFold

After AlphaFold2, the next and predictable evolution was to fully embrace the Transformer architecture and pLMs, which produced methods such as ESMFold [93]. Unlike AlphaFold2, which still heavily depends on large high-quality MSAs for its prediction, ESMFold replaces the MSA input with a pLM-based sequence embedding. This allows ESMFold to predict 3D structures in only a fraction of the time needed by AlphaFold2, though at the cost of a minor loss in model quality. However, for protein sequences with only very few or no homologs at all (resulting in very small or empty MSAs), ESMFold often reached better prediction performances than AlphaFold2.

1.5.3. TmAlphaFold & TMvisDB

Recently, 3D protein structure databases emerged that combine pre-computed AlphaFold2 predictions with membrane protein prediction methods for easy access to membrane protein 3D structures. TmAlphaFold [109] employs a combination of SignalP 6.0 [110], TMDET [39], and CCTOP [63, 64] to filter and detect potential membrane proteins and the boundaries of the membrane bilayer. It currently provides predicted structures and annotations for about 215,000 alpha-helical TMPs. TMvisDB [111] takes a more lightweight approach, pre-filtering the over 200 million structures in AlphaFold DB using TMbed. In total, TMvisDB contains about 44 million alpha-helical TMPs and 2 million beta-barrel TMPs. Though the 3D structures do not explicitly include the membrane boundaries, the TMH and TMB segments predicted by TMbed are highlighted in the 3D structures, making it easy to estimate the membrane boundaries by eye.

1.6. Outline of This Work

This dissertation aims to ease and advance the research of transmembrane proteins (TMP) using state-of-the-art prediction methods based on machine learning. Though TMPs are prevalent in nature [1] and of high interest to both the research community and pharmaceutical industry [22], there is a significant lack of high-quality 3D structures. Over the last three decades, countless statistical and machine learning methods have been developed to predict the topology of TMPs from their primary sequence, in an effort to alleviate the structure gap. In Chapter 2, I introduce TMSEG [65], a method to predict alpha-helical TMPs, their transmembrane helices (TMH) and overall topology, as well as signal peptides (if any). The multi-stage architecture is composed of two random forests (RF), a neural network (NN), and an empirical filter. Leveraging evolutionary information in the form of position-specific scoring matrices (PSSM [72, 77]), TMSEG was able to compete with other state-of-the-art methods of its time. Chapter 3 focuses on PredictProtein [112], one of the oldest web services for protein feature prediction. Running for three decades, users around the world can easily upload their protein sequences and in return get access to various predictions about its potential function and structure [112, 113]. Over time, this included predictions for TMPs by PHDhtm [42–44], PROFtmb [13, 53], and TMSEG. Thus, scientists could use it to scan their proteins for TMPs and further enrich them with annotations including Gene Ontology (GO) [104], secondary structure [89], subcellular location [114], or effect of point mutations [115]. In Chapter 4, I present TMbed [70]. Intended as the successor to TMSEG, it improves upon it in several ways. First, it adds the capability to predict beta-barrel TMPs, making it one of the few methods predicting both types of TMPs at the same time. Next, it utilizes the newest breakthrough in protein feature generation, namely protein language models [89] (pLM). Lastly, it simplified the prediction model architecture and, due to GPU acceleration, is now able to predict entire proteomes within less than an hour. Altogether, TMbed is significantly faster and more accurate than TMSEG, competing with the current state-of-the-art methods. Finally, I will conclude my dissertation with a brief summary in Chapter 5.

2. TMSEG: Novel Prediction of Transmembrane Helices

2.1. Preface

Combining established concepts of previous methods, we developed TMSEG, a prediction method for alpha-helical transmembrane proteins (TMP). First, we decided to use evolutionary information as input, in particular from position-specific scoring matrices (PSSM [72, 77]). This had been proven to increase the prediction performance over only the protein sequence alone [54, 55]. Next, we combined the prediction of transmembrane alpha helices (TMH) with signal peptide prediction, which typically reduces the number of false positive predictions for both of those classes [52]. Finally, we treated TMHs as segments of consecutive residues, rather than predicting the state of each residue individually. This idea was based in the success of hidden Markov models (HMM) in other methods [47–50, 52, 54, 56, 57], though we did go with a different approach.

The architecture of TMSEG is a multi-stage prediction pipeline comprised of two random forests (RF), one neural network (NN), and an empirical filter. In the first stage, a RF predicts the preferences of each residue for three different states: TMH, signal peptide, or other. Those predictions are then smoothed by a median filter in the second stage to remove outliers, and an empirical filter removes all remaining TMHs and signal peptides that are too short. If the protein retains any TMH, it continues to the third stage. Here, a NN adjusts the length and placement of all TMHs and potentially splits very long TMHs. The NN was specifically trained to distinguish between TMH segments that are correctly placed and those that either are non-membrane regions or significantly shifted from their correct position. Thus, treating TMHs as segments of consecutive residues rather than individual ones. The final stage then predicts the inside/outside topology of the non-membrane regions in-between the predicted TMHs.

For the training, we collected TMP data from the Orientations of Proteins in Membranes [5] (OPM) database and the Protein Data Bank of Transmembrane Proteins [33–35] (PDBTM). Further, we took the data set of soluble proteins with and without signal peptides used to train the SignalP4.1 [116] method. We carefully removed redundant sequences to avoid overfitting on a particular protein family. During data processing, we discovered that the annotation from OPM and PDBTM sometimes differed by several residues, i.e. the TMH were shifted. As neither database could be considered more correct than the other, we decided to use only proteins contained in both databases and to train on the annotations of both, hoping that the model will learn to extrapolate the most likely true annotation.

Evaluating TMSEG on an independent test split, it was able to compete with other state-of-the-art methods at the time: MEMSAT3 [55], MEMSAT-SVM [58], and PolyPhobius [54]. It detected $98 \pm 2\%$ of all TMPs in the test set, while only misclassifying $3 \pm 1\%$ of all soluble proteins. For $66 \pm 6\%$ of all TMPs the predicted location of each TMH was within five residues of the annotated position. Especially of note was its rather low false positive rate, i.e., the percentage of soluble proteins incorrectly predicted as TMPs. As the majority of proteins are not TMPs, this means that TMSEG would make significantly fewer mistakes when predicting large quantities of proteins, e.g., whole proteomes or databases. Further, due to the modularity of its architecture, we also tried applying the last two stages (TMH refinement and topology prediction) to the predictions of the other methods. We found that this was able to improve their predictions, demonstrating the usefulness of our segment-based NN.

TMSEG is freely available on GitHub (`https://github.com/Rostlab/TMSEG`) and as part of the PredictProtein [112, 113] (`https://predictprotein.org/`) web service.

Author contribution: I designed and developed the TMSEG method, collected all data sets, and performed all evaluations. Edda Kloppmann and Jonas Reeb co-designing the new evaluation criteria. All authors drafted the manuscript.

2.2. Journal Article: Michael Bernhofer *et al.*, Proteins (2016)

Reference: Bernhofer, M., Kloppmann, E., Reeb, J., and Rost, B. Tmseg: Novel prediction of transmembrane helices. *Proteins*, 84(11):1706–1716, 2016. 10.1002/prot.25155

PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

TMSEG: Novel prediction of transmembrane helices

Michael Bernhofer,[1]* Edda Kloppmann,[1,2] Jonas Reeb,[1] and Burkhard Rost[1,2,3,4]

[1] Department of Informatics & Center for Bioinformatics & Computational Biology – i12, Technische Universität München (TUM), Boltzmannstr. 3, Garching/Munich 85748, Germany

[2] New York Consortium on Membrane Protein Structure, New York Structural Biology Center, New York, New York 10027

[3] Institute of Advanced Study (TUM-IAS), Lichtenbergstr. 2a, Garching/Munich 85748, Germany

[4] Institute for Food and Plant Sciences WZW – Weihenstephan, Alte Akademie 8, Freising, Germany

ABSTRACT

Transmembrane proteins (TMPs) are important drug targets because they are essential for signaling, regulation, and transport. Despite important breakthroughs, experimental structure determination remains challenging for TMPs. Various methods have bridged the gap by predicting transmembrane helices (TMHs), but room for improvement remains. Here, we present TMSEG, a novel method identifying TMPs and accurately predicting their TMHs and their topology. The method combines machine learning with empirical filters. Testing it on a non-redundant dataset of 41 TMPs and 285 soluble proteins, and applying strict performance measures, TMSEG outperformed the state-of-the-art in our hands. TMSEG correctly distinguished helical TMPs from other proteins with a sensitivity of 98 ± 2% and a false positive rate as low as 3 ± 1%. Individual TMHs were predicted with a precision of 87 ± 3% and recall of 84 ± 3%. Furthermore, in 63 ± 6% of helical TMPs the placement of all TMHs and their inside/outside topology was correctly predicted. There are two main features that distinguish TMSEG from other methods. First, the errors in finding all helical TMPs in an organism are significantly reduced. For example, in human this leads to 200 and 1600 fewer misclassifications compared to the second and third best method available, and 4400 fewer mistakes than by a simple hydrophobicity-based method. Second, TMSEG provides an add-on improvement for any existing method to benefit from.

Proteins 2016; 84:1706–1716.

Key words: membrane protein; protein structure prediction; transmembrane helices; α-helical integral membrane protein; transmembrane protein prediction; transmembrane helix prediction.

INTRODUCTION

Transmembrane proteins (TMPs) are involved in numerous essential processes within living organisms such as signaling, regulation, and transport.[1] About 20–30% of all proteins within any organism have been estimated to be TMPs.[2,3] Many TMPs, especially G protein-coupled receptors (GPCRs), are primary drug targets[4] and therefore of high interest.

TMPs cross the membrane bilayer with either transmembrane helices (TMHs) or beta-strands. The latter are found in the outer membrane of Gram-negative bacteria, mitochondria, and chloroplasts. They make up only about 1–2% of all proteins in Gram-negative bacteria.[5] We concentrated on the more common class of helical TMPs and will refer to these as TMPs in the following. TMPs can cross the membrane only once (single-pass) or multiple times (multi-pass). Due to the apolar and hydrophobic environment in the lipid bilayer, most of the amino acids found in TMHs are hydrophobic, and their orientation in the membrane (called TMP topology) can be discerned through Gunnar von Heijne's positive-inside rule.[6,7]

Additional Supporting Information may be found in the online version of this article.

Abbreviations used: 3D, three-dimensional; GPCR, G protein-coupled receptor; NN, (artificial) neural network; OPM, Orientations of Proteins in Membranes; PDB, Protein Data Bank; PDBTM, Protein Data Bank of Transmembrane Proteins; RF, random forest; TMH, transmembrane alpha-helix; TMP, transmembrane protein.

Grant sponsor: Alexander von Humboldt Foundation; Grant sponsor: National Institutes of Health (NIH); Grant number: U54 GM095315.
*Correspondence to: Michael.Bernhofer@mytum.de
Received 22 May 2016; Revised 18 July 2016; Accepted 24 August 2016
Published online 26 August 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/prot.25155

Despite their immense importance, and despite crucial experimental advances,[8–11] <2% of the structures in the Protein Data Bank[12] (PDB) are TMPs.[13–15] As membrane regions are typically not visible in high-resolution structures, TMHs are assigned to PDB structures by expert resources, most prominently the Orientations of Proteins in Membranes[16] (OPM) database and the Protein Data Bank of Transmembrane Proteins[17] (PDBTM).

Recent advances in experimental structure determination have benefited from advanced computational predictions of TMHs from sequence.[8,9] In the last 25 years, many such tools have been developed, ranging from simple algorithms based solely on hydrophobicity scales (e.g., TopPred[18]) to sophisticated uses of hidden Markov models (e.g., TMHMM,[19] HMMTOP,[20] Phobius,[21] and PolyPhobius[22]), neural networks (e.g., PHDhtm,[23,24] and MEMSAT3[25]), and support vector machines (MEMSAT-SVM[26]). Arguably, the most important advance was the incorporation of evolutionary information from sequence profiles or multiple sequence alignments.[23,24] Consequently, almost all methods developed over the last decade are based on evolutionary information. A recent assessment applying strict evaluation measures showed that many methods perform well overall; the best are some recent methods.[27] Here, we show that a few simple ideas improve significantly over the state-of-the-art.

MATERIAL AND METHODS

Dataset TMP166: helical TMPs with known structures

We collected helical TMPs with known structures annotated in OPM[16] and PDBTM[17] (releases 2013_07). Both databases use PDB[12] chain identifiers. We mapped those PDB chains to their UniProtKB[28] protein sequences using SIFTS.[29] We excluded all chimeric PDB chains, model structures, X-ray structures with >8 Å, and those for which some TMH residues did not map gapless to UniProtKB sequences. This gave 1087 PDB chains from 455 PDB structures (379 X-ray and 76 NMR structures).

UniqueProt[30] reduced sequence-redundancy at HVAL > 0 (the HVAL depends on alignment length and the percentage of pairwise sequence identity[31]). At this threshold, no pair of proteins has >20% pairwise sequence identity for alignments of >250 residues (see Rost 1999[32] for precise definitions). The result of this is our final dataset consisting of 166 non-redundant TMPs (called TMP166, Supporting Information Table S1).

As the TMH annotations in OPM and PDBTM differed for some proteins, we associated TMH annotations from both databases with each sequence. The inside/outside topology of the non-transmembrane regions was assigned based on the ATOM coordinates and topology annotation from OPM (cf. Note Supporting Information S1 and Fig. S1). We considered re-entrant regions[33,34] to be non-transmembrane due to their scarcity in the TMP166 dataset (only 15 proteins with one or two re-entry regions each; Supporting Information Table S1).

Dataset SP1441: proteins with and without signal peptides

As signal peptides are often confused with TMHs and vice versa,[27] a second dataset was derived from the SignalP4.1 dataset.[35] This dataset contained UniProtKB sequences of soluble proteins and TMPs with and without signal peptide annotations. Note that these TMPs have no inside/outside topology annotations and many of their TMH annotations are not supported by experimental evidence.

The SignalP4.1 dataset was redundancy reduced twice using UniqueProt. First, all proteins similar to any of those in the TMP166 dataset were removed at HVAL > 0. Second, the remaining proteins were redundancy-filtered at HVAL > 0. The final dataset contained 1441 proteins sequences (299 TMPs and 1142 soluble proteins, called SP1441; Supporting Information Table S2). About 477 of those had signal peptide annotations (25 TMPs and 452 soluble proteins).

Splitting the datasets

We split the combined TMP166 and SP1441 dataset into four subsets. We partitioned them in a way that all subsets have approximately the same distributions with respect to the number of soluble proteins and TMPs, protein sequences with and without signal peptides, and sequence lengths (Supporting Information Fig. S2).

We used the first three subsets to develop TMSEG in a three-fold cross-validation approach (cf. TMSEG training). The fourth split, the independent test set called BlindTest, was used only for the final performance evaluation, i.e., no parameter was optimized on that set. The BlindTest dataset contained 41 TMPs (from TMP166) with known structure and TMH annotations from OPM and PDBTM, and 285 soluble proteins from the SP1441 dataset. The 74 TMPs from the fourth split of SP1441 (Supporting Information Table S2) were not included in the BlindTest dataset, because they lack sufficient experimental annotations. However, we used them for the signal peptide prediction performance analysis, as we did not have curated signal peptide annotations for the TMPs from OPM and PDBTM.

Human proteome

We retrieved the human proteome, 20,196 protein sequences, from UniProtKB/Swiss-Prot (release 2015_03). We applied our TMSEG algorithm to the whole proteome to provide a summary of its TMP composition and to estimate run time.

Table I
Evaluation Measures

Measurement	Formula	Description
Precision (%)	$100 * \frac{\#\text{ of correctly predicted TMHs}}{\#\text{ of predicted TMHs}}$	Precision of TMH prediction
Recall (%)	$100 * \frac{\#\text{ of correctly predicted TMHs}}{\#\text{ of observed TMHs}}$	Recall of TMH prediction
Q_{ok} (%)	$\frac{100}{N} * \sum_{i=1}^{N} x_i;\ x_i = \begin{cases} 1, & if\ p_i = r_i = 100\% \\ 0, & else \end{cases}$	Percentage of TMPs with correct TMH placement
Q_{top} (%)	$\frac{100}{N} * \sum_{i=1}^{N} y_i;\ y_i = \begin{cases} 1, & if\ p_i = r_i = t_i = 100\% \\ 0, & else \end{cases}$	Percentage of TMPs with correct TMH placement and inside/outside topology
FPR (%)	$100 * \frac{\#\text{ of incorrectly predicted TMPs}}{\#\text{ of soluble proteins}}$	False positive rate of TMP prediction
Sensitivity (%)	$100 * \frac{\#\text{ of correctly predicted TMPs}}{\#\text{ of observed TMPs}}$	Sensitivity of TMP prediction

Listed are the evaluations measures used and how they were calculated. Precision and recall for the performance evaluation of the TMH prediction were computed by combining all TMHs within the dataset (i.e., not averaged over each protein). Q_{ok} and Q_{top} were calculated based on all TMPs, where N was the number of TMPs in the dataset, p_i and r_i were the TMH precision and recall for protein i within the dataset, and $t_i = 100\%$ indicated a correctly predicted N-terminal inside/outside topology for protein i.

Dataset New12

Our original datasets had been based on the PDB release from July 2013, when this work began. Shortly before submission of the work in February 2016, that is, 32 months later, we retrieved all TMPs added to OPM and PDBTM since July 2013. We removed all TMPs similar (HVAL > 0) to proteins in datasets used previously (TMP166 and SP1441). Testing the pairwise similarity of the remaining TMPs we found that two pairs were similar (HVAL > 0), but we decided to keep them due to their low HVAL. This resulted in 12 new TMPs (New12 dataset, Supporting Information Table S3) we used for additional testing. Although the statistical power of such a small set is very limited, these 12 constitute the entire addition of completely new structures from 2013/07 to 2016/02. Further, these or structurally related TMPs have most likely not been used to develop any method used for comparison.

Evaluation

As per-protein scores (correct classification as TMP or non-TMP), we compiled the sensitivity (percentage of observed TMPs predicted as TMPs) and the false positive rate (FPR: percentage of soluble proteins predicted as TMPs, Table I). As per-TMH scores (correct identification and placement of TMHs), we compiled the precision (percentage of predicted TMHs that are correct), recall (percentage of observed TMHs predicted as TMHs), Q_{ok} and Q_{top}. Q_{ok} is the percentage of TMPs for which all TMHs are correctly predicted (Table I). Q_{top} requires in addition to Q_{ok} correct topology predictions (in/out: Table I). To resolve conflicts between OPM and PDBTM annotations, we chose whichever fit the prediction best. Note that while sensitivity and recall have the same formula, we used sensitivity in conjunction with TMPs and recall with TMHs to better distinguish between those scores in the text.

Each TMH was considered correctly predicted, if predicted and observed TMH ends were within five residues (Supporting Information Fig. S3), and if predicted and observed TMH overlapped by at least half of the length of the longer of the two helices. These two criteria are more stringent than those that have commonly been used (typically: overlap >3–5 residues anywhere between observed and predicted TMH[36]) and have recently led to re-evaluating TMH prediction methods.[27] None of our major conclusions changed upon applying values slightly different than five residues for the maximum allowed discrepancy between predicted and observed TMH ends (data not shown).

Error rates for the evaluation measures were estimated by bootstrapping,[37] i.e., by re-sampling the population of proteins used for the evaluation 1000 times and calculating the sample standard deviation. Each of these sample populations contained 60% of the original proteins (picked randomly without replacement).

State-of-the-art methods

We compared TMSEG to the best methods,[27] namely to PolyPhobius,[22] MEMSAT3,[25] and MEMSAT-SVM.[26] Like TMSEG, these methods also use evolutionary information to predict TMPs: MEMSAT3 and MEMSAT-SVM automatically generate position-specific scoring matrices (PSSMs) with PSI-BLAST, while PolyPhobius generates multiple sequence alignments (MSAs). To ensure equal conditions for all methods we ran them on our local machines and used the UniProt Reference Cluster with

90% sequence identity (UniRef90, release 2015_03) as the homology search database, i.e., to generate the MSAs or PSSMs. While we used proteins completely unknown to TMSEG to assess its performance, some of the proteins used in our assessment might have been used to develop PolyPhobius, MEMSAT3, or MEMSAT-SVM. In this sense, our assessment was likely to over-estimate their performance, in particular with respect to TMSEG.

Baseline performance

We also compared all methods to a simple baseline predictor similar to TopPred[18]: for all possible segments of 21 consecutive residues, we summed the Eisenberg-hydrophobicity[38] (EisenbergSum, Supporting Information Table S4). All non-overlapping segments with EisenbergSum ≥ 4 were predicted as TMHs, starting with the segments with the highest sum. The inside/outside topology was predicted based on the difference between arginine and lysine residues on either side of the TMHs, i.e., applying Gunnar von Heijne's positive-inside rule.[6,7]

TMSEG input/output

TMSEG needs two input files to successfully run a prediction: a FASTA file with the protein sequence and a PSI-BLAST PSSM file for the input protein. The PSSM file is mandatory and used to include homology-based features that greatly increase the prediction accuracy.

Combining evolutionary information (e.g., PSSMs and MSAs) with machine learning has been the most important improvement in protein prediction and is commonly used in TMH and secondary structure prediction.[24,27,39,40] TMSEG incorporates evolutionary information through PSI-BLAST profiles[41] generated from UniRef90 (release 2015_03). We used two sets of profiles: a training set with a stringent E-value cutoff of 10^{-5} and five iterations for creating the profile, as well as a test set with a less strict E-value cutoff of 10^{-3} and three iterations. We deactivated PSI-BLAST's low-complexity filter and enabled the option to calculate local optimal Smith-Waterman alignments in order to generate longer and more accurate alignments.

In addition, we used biophysical properties (charge, hydrophobicity, polarity; Supporting Information Table S4) and the overall amino acid composition. These features were calculated twice for each residue: once for all substitutions with a positive PSSM score and once based on all substitutions with a negative score.

The standard output gives a brief summary of the positions of the TMHs and signal peptide (if any) and the inside/outside topology. In addition, a raw output is available that also contains the unmodified output probabilities of the machine-learning tools.

TMSEG algorithm

TMSEG combines several machine-learning tools and empirical filters. The machine-learning algorithms used are two random forests (RFs) and one neural network (NN), both of which are implementations from the WEKA Java package.[42] The output of these algorithms is further processed with empirically determined filters and thresholds. The TMSEG algorithm executes four separate steps (Fig. 1):

Step 1: initial per-residue prediction

An RF detects TMHs from the input sequence. This RF slides a window of 19 consecutive residues through the protein sequence, predicting whether or not the central residue in the window is in a TMH, signal peptide, or non-TM region, i.e., the probability of each residue for each state is calculated based on the residue itself and the nine residues left and right of it. For each of the 19 residue positions, we compute the PSSM profile. For the central nine residues in the window, we also compute the average Kyte-Doolittle[43] hydrophobicity, and the percentage of hydrophobic, charged, and polar residues (Supporting Information Table S4).

In addition to these local features, we compile global features: the distance of the residue to the N- and C-terminus, the length of the protein sequence, and the global amino acid composition. The RF assigns three values to each residue corresponding to the probability to be in a TMH, a signal peptide, or a non-TM region. Runtime is decreased by multiplication of the probabilities by 1000 and transformation into integers.

Step 2: per-protein filter: TMP or soluble

The per-residue scores are filtered empirically. First to reduce short peaks of one or two residues, all per-residue scores are smoothed by compiling the median score over five consecutive residues and assigning it to the center residue. Next, each residue is assigned to the state with the highest score (TMH, signal peptide, or non-TM). To prevent over-prediction due to the under-sampling of signal peptide residues, we applied a penalty of 185 (that is, 18.5%) to non-TM and 60 (that is, 6%) to TMH residues. These penalties were optimized during cross-training to best balance over- and under-prediction. Finally, TMHs shorter than seven residues are changed into non-TM regions. If a signal peptide of at least four consecutive residues is identified within the first 40 N-terminal residues ending in residue at position i, TMSEG predicts a signal peptide from residue 1 to residue i ($i \leq 40$). Signal peptide predictions outside the first 40 residues ($i > 40$) are changed into non-TM, but do not invalidate signal peptides inside the first 40 residues.

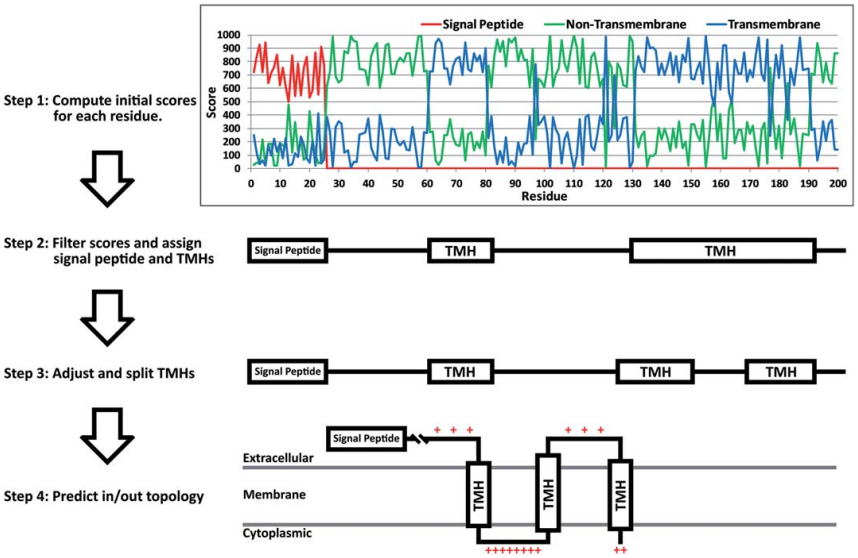


Figure 1

TMSEG algorithm. The new method TMSEG has four steps of machine learning and optimization. Step 1: A random forest (RF) assigns a score to each residue for the three states transmembrane helix (TMH), signal peptide, and non-TM region. Step 2: The previous scores are smoothed (median over 5 residues), all residues are assigned to the state with the highest score, and short segments are removed. Step 3: A segment-based neural network (NN) adjusts the exact position of predicted TMHs, and their length, sometimes splitting TMHs, sometimes shifting, extending, or compressing them. Step 4: The inside/outside topology is predicted by another RF.

Initial predictions with fewer than four consecutive residues are changed into non-TM.

Step 3: refinement of TMHs

In the third step an NN corrects the predicted TMHs. In contrast to the standard sliding window approach of the RF in Step 1, here we introduced a segment-based solution that used as input the following averages over the predicted TMHs: length of predicted TMH, amino acid composition, average hydrophobicity, as well as the percentages of hydrophobic and charged residues. The output of the NN is the predicted probability for the segment to be a TMH. Based on this probability, the predicted TMHs from Step 2 are adjusted.

First, TMHs ≥ 35 residues are split into two TMHs with at least 17 residues, if these two TMHs increase the overall probability. The minimum length of 35 residues for splitting long TMHs and of 17 residues for the resulting two TMHs were empirically chosen based on the overall performance during cross-training. Second, the start and end positions for each TMH are adjusted by shifting them by up to three residues in either direction. Shifts are accepted if they increase the overall probability. The maximum endpoint adjustment by three residues was empirically chosen based on the overall performance during cross-training. In addition, the relatively long minimum TMH lengths to allow splitting and the relatively small shift of maximally three residues of the TMH ends allow TMSEG to maintain a short runtime.

Step 4: topology prediction

Another RF predicts the inside/outside topology of the TMP, i.e., in which direction the TMHs cross the membrane. During this step the non-transmembrane regions are assigned to inside (e.g., cytoplasmic side of the membrane) or outside. This prediction is made for the entire protein. For each TMH, we consider up to 15 residues before and after the TMH, and eight residues at the TMH start and end (for TMHs < 16 these residues overlap). As all predicted TMHs are assumed to cross the membrane, the in/out assignment is switched after each TMH. For each side, we compute as input to the RF the amino acid composition, the percentage of positively charged residues (we consider all arginine and lysine residues), and the absolute difference of positively charged residues between the two sides. Based on the RF output, one side is assigned to be inside (e.g., cytoplasmic), the other to be outside. Residues immediately after predicted signal peptides are assigned to outside (non-cytoplasmic) and all

Table II
Per-Protein Distinction Between Helical TMPs and Other Proteins

Method	TMP sensitivity	TMP FPR	Topology correct	Misclassified in human	More mistakes than TMSEG in human
TMSEG	98 ± 2	3 ± 1	93 ± 4	558	-
PolyPhobius[22]	100 ± 0	5 ± 1	78 ± 7	770	212
MEMSAT3[25]	100 ± 0	28 ± 2	93 ± 4	4313	3755
MEMSAT-SVM[26]	98 ± 2	14 ± 2	88 ± 5	2253	1695
Baseline	95 ± 3	31 ± 2	75 ± 7	5015	4457

Results are provided for all 41 TMPs and 285 soluble proteins in the BlindTest dataset. Error rates are the sample standard deviation based on bootstrapping (cf. Methods). Listed are the *TMP sensitivity* (percentage of correctly predicted helical TMPs), the *TMP FPR* (percentage of non-TMP proteins incorrectly predicted as TMP), *Topology correct* (percentage of proteins for which the topology (inside/outside) was correctly predicted; this differs from Q_{top} which requires topology and all TMHs to be predicted correctly), *Misclassified in human* (estimates the number of proteins misclassfied for the entire human proteome), and *More mistakes than TMSEG in human* (estimates the number of proteins misclassfied more by the method than by TMSEG). The estimates for the human proteome are based on two assumptions: (i) the error estimates on the BlindTest dataset hold true for the human proteome, (ii) the human proteome has 20,196 proteins, 4791 of which are TMPs (cf. Results section "*Application to the human proteome*").

consecutive segments are assigned accordingly without any further prediction.

TMSEG training

To reduce the risk of over-fitting, we split our combined TMP166 and SP1441 datasets into four even splits (cf. Supporting Information Tables S1 and S2). Note that the TMPs from the SP1441 dataset were used to train the random forest in the initial prediction (step 1) as they contain signal peptide annotations. They are, however, not used for the neural network (step 3) or the random forest in step 4, since they have no inside/outside topology annotations and many of their TMH annotations are not supported by experimental evidence.

The first of three splits was used to train, the second to cross-train, i.e., to optimize all other free parameters (e.g., the minimum TMH length), and the last to evaluate performance (test). This procedure was repeated three times, such that each protein had been used exactly once for training, cross-training and testing. The final parameters were frozen according to the overall best performance for all three rotations (on the test set). Given the frozen parameters, we applied the final method to the fourth split, the BlindTest dataset, which had not been used before.

Our careful four-fold split leading to three-fold development (each with training, cross-training, and testing), provided a double protection against overestimating performance. We decided about every detail in the final method before using the BlindTest dataset to evaluate TMSEG as presented here. Many developers use a two-fold split (training/testing), more careful ones the three-fold split (training/cross-training/testing), while the fourth split is occasionally introduced through pre-release data[39] like the New12 dataset that we generated.

RESULTS AND DISCUSSION

The novel TMSEG method introduced here distinguishes between proteins with transmembrane helices (TMHs) and soluble proteins. For all helical transmembrane proteins (TMPs), it predicts the placement of the TMHs, and their orientation in the membrane, i.e., their inside/outside topology. We established sustained performance through cross-validation with two levels of blind testing. We compared our new methods to others, including the best at predicting TMPs,[27] namely PolyPhobius[22] and MEMSAT-SVM.[26] Furthermore, we analyzed MEMSAT3[25] because it excels at the inside/outside topology prediction,[44] and SignalP4.1 as the leading method for signal peptide identification.[35] In addition, we compared to a simple hydrophobicity-based prediction similar to TopPred.[18]

Outstanding per-protein distinction between TMPs and other proteins

TMSEG correctly identified 40 of the 41 TMPs in the BlindTest dataset (98 ± 2% sensitivity) and incorrectly predicted 8 of 285 soluble proteins as TMPs (3 ± 1% false positive rate: FPR). TMSEG performed similar to PolyPhobius (100% sensitivity and 5 ± 1% FPR) and significantly better than MEMSAT3 and MEMSAT-SVM (Table II).

Although signal peptides can be confused with TMHs due to the similarity of their signal, only one of the 8 mistakes of predicting soluble proteins as TMPs originated from incorrectly predicting a signal peptide as a TMH. This shows that training on a dataset containing signal peptides helped significantly to reduce false positive predictions. PolyPhobius, which also includes a sophisticated signal peptide prediction, did not confuse any signal peptides with TMHs. However, MEMSAT-SVM, MEMSAT3, and the Baseline predictor had 13, 41, and 69 predicted TMHs, respectively, that overlapped by at least half their length with annotated signal peptides. Overall, TMSEG was able to reliably detect signal peptides and to not predict them as TMHs (Supporting Information Table S5).

We used the 74 TMPs from the fourth subset of the SP1441 dataset (cf. Supporting Information Table S2) to further test the prediction of signal peptides and TMHs. For these proteins, TMSEG and PolyPhobius incorrectly predicted several single-pass TMPs as soluble proteins, because they confused their TMHs near the N-terminus with signal peptides (Supporting Information Table S5). This trend did not occur with the TMPs from the TMP166 dataset (evident by their high sensitivity values; Table II). An explanation might be that TMPs with TMHs within the first 40 residues are more prevalent in the SP1441 dataset, which makes this misclassification more likely to happen. Although these misclassification rates would lower our previous sensitivity estimates for TMSEG and PolyPhobius (at least for single-pass TMPs with their TMH near the N-terminus), we hesitate to generalize the results to everyday applicability since the SP1441 dataset is biased (it was generated to develop the signal peptide predictor SignalP4.1) and contains many TMPs with a TMH near the N-terminus. Further, only 2 of the 9 TMHs that were incorrectly predicted as SPs had experimental evidence.

While all methods reached high sensitivity, they differed vastly in their false positive rates, i.e., soluble proteins incorrectly considered to contain TMHs (Table II). By translating the error rates, the number of proteins that would be misclassified in the entire human proteome can be estimated using two reasonable assumptions: (i) the error estimates for all methods based on the 326 non-redundant proteins (41 TMPs and 285 soluble proteins) in the BlindTest dataset hold true for the (redundant) human proteome, (ii) the human proteome has 20,196 proteins and 4791 of those are TMPs (cf. Section below "*Application to the human proteome*"). Under these assumptions, TMSEG achieves 97% per-protein accuracy and misclassifies only about 558 human proteins. The second best method, PolyPhobius, makes 770 mistakes (212 more than TMSEG) and MEMSAT-SVM as the third best method already misclassifies 2253 proteins (1695 more than TMSEG, Table II). In fact, TMSEG is almost 8.8-times superior to the Baseline predictor, PolyPhobius over 6.5-times better, and MEMSAT-SVM 2.2-times better than the Baseline predictor (Supporting Information Table S6).

Best overall per-TMH prediction

Overall, TMSEG achieved a sustained level of precision ($87 \pm 3\%$) and recall ($84 \pm 3\%$) for the TMHs, that is, $87 \pm 3\%$ of all predicted TMHs were at the correct position and $84 \pm 3\%$ of all observed TMHs had been accurately predicted [Supporting Information Fig. S4(A,B)]. These values were second to no other method, however, only slightly above the second best method MEMSAT-SVM ($85 \pm 3\%$ precision at $83 \pm 3\%$ recall). All other methods had scores below 80%. For $66 \pm 6\%$ of all TMPs, TMSEG predicted all observed TMHs at their correct positions, i.e., $Q_{ok} = 66 \pm 6\%$ (Fig. 2). MEMSAT-SVM followed as second best with $Q_{ok} = 61 \pm 7\%$ (Fig. 2). Nevertheless, given the small datasets, the top performance of TMSEG remained within one standard deviation of all compared methods, except the baseline hydrophobicity prediction (Fig. 2: error bars).

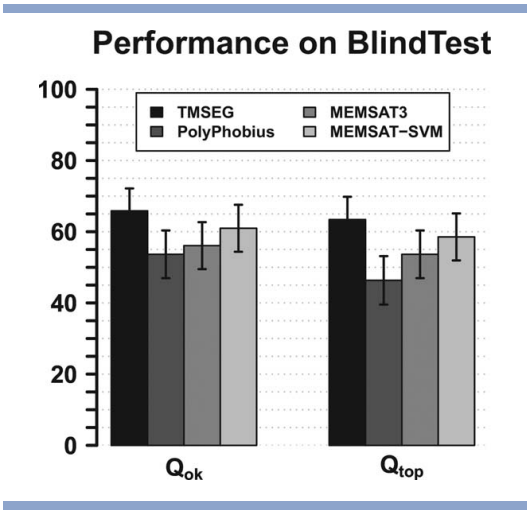


Figure 2

TMSEG compared favorably to state-of-the-art. Results are provided for all 41 TMPs in the BlindTest dataset. Error bars are the sample standard deviation based on bootstrapping (cf. Methods). Shown is on the left the percentage of proteins for which all TMHs were predicted correctly (Q_{ok}, Table I) and on the right the percentage of proteins with correctly predicted TMHs and inside/outside topology (Q_{top}, Table I; note that $Q_{ok} \geq Q_{top}$ by definition).

When comparing the performance on TMP subsets based on the number of TMHs, the performance got worse the more TMHs a protein had [Supporting Information Fig. S4(C,D)]. This might be misunderstood to imply that prediction methods perform better in placing the TMHs in single-pass TMPs than in, e.g., GPCRs (with 7 TMHs). However, this simple numerical comparison ignores the difference in the difficulty of the task: The Baseline predictor reached a high value in Q_{ok} for single-pass TMPs, but failed to predict all TMHs correctly for any TMP with >5 TMHs [Supporting Information Fig. S4(C)]. In fact, when we simply compiled performance for the subset of proteins for which the Baseline predictor failed, we found similar values for proteins with one TMH, those with 2–5, and those with >5 TMHs (Supporting Information Fig. S5).

In contrast, it surprised us that even for the trivial cases, i.e., those for which the Baseline predictor had all TMHs correct, the more advanced methods failed for some of them. This suggests that the large number of different features used by the more advanced methods sometimes interfere with and obscure a strong

hydrophobicity signal. Indeed, only 11 of the 19 trivial TMPs were correctly predicted by all four other methods. However, TMSEG still performed best with $Q_{ok} = 89 \pm 6\%$, followed by MEMSAT3 and MEMSAT-SVM with $Q_{ok} = 84 \pm 7\%$ (data not shown).

Best inside/outside topology prediction

TMSEG and MEMSAT3 correctly placed the N-terminus as inside (e.g., cytoplasmic) or outside (e.g., extracellular), i.e., correctly predicted the topology, for $93 \pm 4\%$ of all TMPs (Table II). When taking into account the global topology and correct TMH placement (i.e., Q_{top}), TMSEG performed better than all other methods reaching $Q_{top} = 63 \pm 6\%$ (Fig. 2). This is five percentage points higher than the second best method, MEMSAT-SVM (albeit still within one standard deviation). Most advanced methods predicted the topology correctly for almost all proteins for which they correctly predicted all TMHs (Q_{top} almost identical to Q_{ok} for all methods, except for the Baseline predictor in Fig. 2).

Application to the human proteome

We applied TMSEG to predict all helical TMPs in the human proteome (20,196 proteins from UniProtKB/Swiss-Prot). TMSEG predicted a total of 5157 TMPs, almost half of these (2300 = 45%) were predicted with one TMH. Given the sensitivity and false positive rate of TMSEG (98 ± 2 and $3 \pm 1\%$, respectively; Table II), we estimate that 462 TMPs were incorrectly predicted (over-predicted) and 96 were missed (under-predicted). In total, we thus misclassified 558 proteins, and our corrected estimate was that humans have about 4791 TMPs, i.e., about 24% of all proteins cross the membrane. While TMSEG misclassified about 558 human proteins, the mistake in the estimate of this percentage appeared to be less than a permille, that is, $\pm 0.01\%$. However, our error estimate might be too simplistic due to the high number of single-pass TMPs for which the error rates are much higher than for proteins with more TMPs.

Confirming previous observations,[2,3] we also observed two peaks of predicted TMPs for proteins with 7 TMHs (819 proteins) and 12 TMHs (189 proteins). These likely represent G protein-coupled receptors (GPCRs) and transporter proteins. Applying UniqueProt to the 5157 predicted TMPs, we found around 500 non-redundant TMPs of which 320 are single-pass TMPs.

Latest experimental structures confirmed our estimates

The 12 new TMPs (New12 dataset) that have recently been added to the PDB constituted the only dataset with truly identical conditions for all methods assessed. The New12 dataset allowed us to confirm the outstanding performance of our new method TMSEG. TMSEG and PolyPhobius correctly identified 10 of the 12 TMPs ($83 \pm 10\%$ sensitivity), while MEMSAT3, MEMSAT-SVM, and the Baseline predictor identified 11 ($92 \pm 7\%$ sensitivity). However, TMSEG correctly predicted every TMH of those 10 TMPs, resulting in a $Q_{ok} = 83 \pm 10\%$, compared to $Q_{ok} = 58 \pm 13\%$ for PolyPhobius, MEMSAT3, and MEMSAT-SVM (Baseline predictor $Q_{ok} = 50 \pm 13\%$). TMSEG also performed best taking into account the topology prediction and reached $Q_{top} = 66 \pm 12\%$, compared to a $Q_{top} = 58 \pm 13\%$ for MEMSAT3 and MEMSAT-SVM, and $Q_{top} = 50 \pm 13\%$ for PolyPhobius and the Baseline predictor.

Comparisons complicated by small datasets

The two small datasets available for evaluation (BlindTest with 41 TMPs and New12 with 12 TMPs) implied high standard errors for many performance estimates. Especially standard errors for the TMH-segment based scores are so high (up to 16 percentage points, Supporting Information Fig. S4) that comparisons between methods hardly provide statistically significant differences on the TMH-segment level. Nevertheless, TMSEG seemed to perform on par with any existing method. Note that the differences in the distinction between helical TMPs and other proteins in the BlindTest dataset were statistically significant even in considering TMSEG as slightly better than the second best PolyPhobius (Table II).

Further, we could not use a single gold standard, because OPM and PDBTM differed in their TMH annotations: comparing the OPM annotations to the PDBTM annotations (that is, "predicting" one with the other) yielded $Q_{ok} = 56 \pm 7\%$. In other words, if we considered one of those experiment-based annotations as the prediction of the other, the average performance would be similar to that of TMSEG and the other methods. When using only OPM or PDBTM annotations to evaluate the prediction performance, TMSEG still performed excellently (Supporting Information Fig. S6). However, this was also the only comparison in which one other method reached a numerically higher value for a dataset than TMSEG, namely MEMSAT-SVM on the PDBTM annotations. Overall, all predictions agreed more with OPM than with PDBTM annotations (Supporting Information Fig. S6).

Performance best with diverse alignments

TMSEG strongly depends on the evolutionary information taken from PSI-BLAST PSSMs. We recommend using a sufficiently large search database (e.g., UniRef90) to generate the PSSMs. Additionally, redundancy reduction might help (e.g., at 90% pairwise sequence identity as in UniRef90).

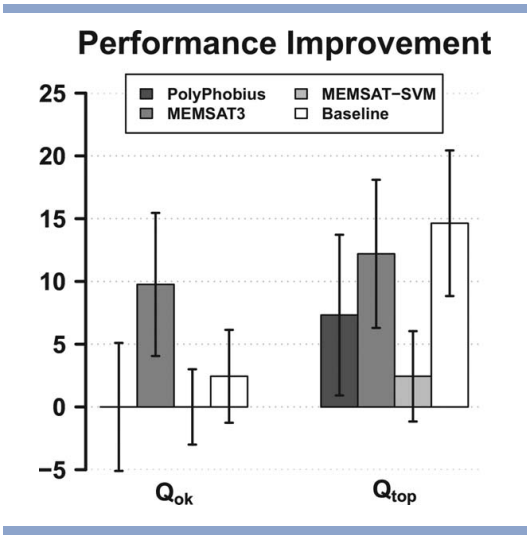


Figure 3

TMSEG applied to refine other methods. The TMSEG algorithm iteratively refines performance through four consecutive steps. Here, we applied Steps 3 and 4 as post-filters to other methods (dataset and error bars as in Fig. 2). Given is the improvement of Q_{ok} and Q_{top} (cf. Table I for definitions) of the prediction method by applying TMSEG, i.e., Q(method + TMSEG) – Q(method). Note that PolyPhobius (first bar on the left) and MEMSAT-SVM (third bar on the left) showed, on average, no improvement in Q_{ok}.

Alignments built from smaller search-databases (e.g., UniRef50 and Swiss-Prot) only slightly lowered the per-protein performance: the sensitivity never dropped below $90 \pm 4\%$, while the false positive rate remained at or below $3 \pm 1\%$. However, the TMH-based precision and recall values dropped substantially (Supporting Information Fig. S7). Thus, for sequences that produce no PSI-BLAST hits, we recommend using a larger search database or—in the rare case that the protein is a true singleton—a method that is independent of evolutionary information, e.g., Phobius.[21,27]

Re-entrant membrane helices not predicted correctly

Our dataset contained only few re-entrant helices, insufficient to learn their prediction (Supporting Information Table S1). Therefore, we considered re-entrant helices as non-TM during training to avoid later interference with the inside/outside topology prediction. Due to the lack of data, we could not reliably assess how well TMSEG distinguishes TMHs from re-entrant membrane helices: The BlindTest dataset included only seven re-entrant regions (OPM and PDBTM annotations combined). TMSEG incorrectly predicted five of seven as TMHs; two of these five were predicted as two separate TMHs; thus, the overall inside/outside topology was not influenced. MEMSAT-SVM, the only tested method that predicts re-entrant helices, identified five of the seven as re-entrant, predicted one as a TMH, and missed the last. When considering re-entrant regions as TMHs, Q_{ok} remained the same for TMSEG and PolyPhobius and dropped by 2–5 percentage points for MEMSAT-SVM, MEMSAT3, and the Baseline predictor.

TMSEG easily combined with other methods

Due to the modularity of TMSEG (i.e., its four separate steps, Fig. 1), it can be used to refine other methods. This includes the adjustment of the TMHs as well as the inside/outside topology prediction. We used the TMH predictions of the reference methods, and applied Steps 3 and 4 of TMSEG to their prediction (Fig. 2). Applying TMSEG as refinement improved the performance for most methods (Fig. 3; Supporting Information Fig. S8). While the improvement was small for the TMH placement (Q_{ok}), TMSEG improved most methods by over eight percentage points in Q_{top} (correct TMHs and topology).

Runtime estimation

We estimated the runtime by applying TMSEG to the human proteome (20,196 proteins). As the time to run PSI-BLAST differs depending on the database size, we decided to use pre-computed PSSMs to measure only the time needed by TMSEG. Given those PSI-BLAST profiles, the prediction for the entire human proteome took about 90 min (Intel Core i7-3632QM 2.2 GHz, 8GB RAM; no multithreading), which corresponds to three to four protein sequences per second.

CONCLUSION

In our hands, our new method TMSEG almost always outperformed existing state-of-the-art prediction methods (Table II, Fig. 2). However, due to the small datasets, many improvements on the per-TMH level remained too small for the large margin of statistical significance (standard errors up to 16 percentage points, Supporting Information Fig. S4). Most importantly, TMSEG achieved the significantly best per-protein classification in the distinction between helical TMPs and all other proteins. For instance, for the prediction of all human proteins, this implied about 558 incorrectly predicted proteins. This number might appear high; however, no method tested reached such a low level, e.g., PolyPhobius misclassified about 200 more proteins than TMSEG and MEMSAT-SVM fared about four times worse (corresponding to >2000 incorrect predictions).

The highest per-protein performance resulted from a combined prediction of TMHs, non-TM regions, and signal peptides. In order to predict re-entrant helices,

another state would have to be introduced; as is, TMSEG predicted five of seven re-entrant helices in our dataset as TMHs. The sustained high levels of per-segment predictions resulted from our new segment-focused algorithm. Another major advantage of our new concept is that it can be used to improve the predictions of most other TMH prediction methods.

Availability and speed

Other than its top performance, using TMSEG may also be recommended due to its speed and because it might help to improve over the method that you run locally. The method is easily and freely available: online through the PredictProtein[45] webserver (www.predictprotein.org), and as standalone Debian package from the Rostlab Debian repository (www.rostlab.org/owiki) and GitHub (www.github.com/Rostlab/TMSEG). A tutorial on how to use PSI-BLAST and TMSEG can be found in the Rostlab Wiki (www.rostlab.org/owiki/index.php/TMSEG).

ACKNOWLEDGMENTS

Thanks to Tim Karl for technical and to Inga Weise (both TUM) for administrative assistance. Thanks to all authors who made their methods openly available and provided us with versions to run on our own machines. Last but not least, thanks to all who practice open science and deposit their data into public databases and those who maintain these excellent databases.

REFERENCES

1. von Heijne G. The membrane protein universe: what's out there and why bother? J Intern Med 2007;261:543–557.
2. Liu J, Rost B. Comparing function and structure between entire proteomes. Protein Sci 2001;10:1970–1979.
3. Fagerberg L, Jonasson K, von Heijne G, Uhlen M, Berglund L. Prediction of the human membrane proteome. Proteomics 2010;10:1141–1149.
4. Overington JP, Al-Lazikani B, Hopkins AL. How many drug targets are there? Nat Rev Drug Discov 2006;5:993–996.
5. Bigelow HR, Petrey DS, Liu J, Przybylski D, Rost B. Predicting transmembrane beta-barrels in proteomes. Nucleic Acids Res 2004;32:2566–2577.
6. von Heijne G. The distribution of positively charged residues in bacterial inner membrane proteins correlates with the trans-membrane topology. EMBO J 1986;5:3021–3027.
7. von Heijne G, Gavel Y. Topogenic signals in integral membrane proteins. Eur J Biochem 1988;174:671–678.
8. Punta M, Love J, Handelman S, Hunt JF, Shapiro L, Hendrickson WA, Rost B. Structural genomics target selection for the New York consortium on membrane protein structure. J Struct Funct Genomics 2009;10:255–268.
9. Love J, Mancia F, Shapiro L, Punta M, Rost B, Girvin M, Wang DN, Zhou M, Hunt JF, Szyperski T, Gouaux E, MacKinnon R, McDermott A, Honig B, Inouye M, Montelione G, Hendrickson WA. The New York Consortium on Membrane Protein Structure (NYCOMPS): a high-throughput platform for structural genomics of integral membrane proteins. J Struct Funct Genomics 2010;11:191–199.
10. Caffrey M. A comprehensive review of the lipid cubic phase or in meso method for crystallizing membrane and soluble proteins and complexes. Acta Crystallogr F Struct Biol Commun 2015;71:3–18.
11. Moraes I, Evans G, Sanchez-Weatherby J, Newstead S, Stewart PD. Membrane protein structure determination - the next generation. Biochim Biophys Acta 2014;1838:78–87.
12. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res 2000;28:235–242.
13. Kloppmann E, Punta M, Rost B. Structural genomics plucks high-hanging membrane proteins. Curr Opin Struct Biol 2012;22:326–332.
14. White SH. Biophysical dissection of membrane proteins. Nature 2009;459:344–346.
15. White SH. The progress of membrane protein structure determination. Protein Sci 2004;13:1948–1949.
16. Lomize MA, Lomize AL, Pogozheva ID, Mosberg HI. OPM: orientations of proteins in membranes database. Bioinformatics 2006;22:623–625.
17. Kozma D, Simon I, Tusnady GE. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. Nucleic Acids Res 2013;41:D524–529.
18. von Heijne G. Membrane protein structure prediction. Hydrophobicity analysis and the positive-inside rule. J Mol Biol 1992;225:487–494.
19. Krogh A, Larsson B, von Heijne G, Sonnhammer EL. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J Mol Biol 2001;305:567–580.
20. Tusnady GE, Simon I. The HMMTOP transmembrane topology prediction server. Bioinformatics 2001;17:849–850.
21. Käll L, Krogh A, Sonnhammer EL. A combined transmembrane topology and signal peptide prediction method. J Mol Biol 2004;338:1027–1036.
22. Käll L, Krogh A, Sonnhammer EL. An HMM posterior decoder for sequence feature prediction that includes homology information. Bioinformatics 2005;21: i251–257.
23. Rost B, Casadio R, Fariselli P, Sander C. Transmembrane helices predicted at 95% accuracy. Protein Sci 1995;4:521–533.
24. Rost B, Casadio R, Fariselli P. Refining neural network predictions for helical transmembrane proteins by dynamic programming. Proc Int Conf Intell Syst Mol Biol 1996;4:192–200.
25. Jones DT. Improving the accuracy of transmembrane protein topology prediction using evolutionary information. Bioinformatics 2007;23:538–544.
26. Nugent T, Jones DT. Transmembrane protein topology prediction using support vector machines. BMC Bioinformatics 2009;10:159
27. Reeb J, Kloppmann E, Bernhofer M, Rost B. Evaluation of transmembrane helix predictions in 2014. Proteins 2015;83:473–484.
28. UniProt C. UniProt: a hub for protein information. Nucleic Acids Res 2015;43:D204–212.
29. Velankar S, Dana JM, Jacobsen J, van Ginkel G, Gane PJ, Luo J, Oldfield TJ, O'Donovan C, Martin MJ, Kleywegt GJ. SIFTS: Structure Integration with Function, Taxonomy and Sequences resource. Nucleic Acids Res 2013;41:D483–489.
30. Mika S, Rost B. UniqueProt: Creating representative protein sequence sets. Nucleic Acids Res 2003;31:3789–3791.
31. Sander C, Schneider R. Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins 1991;9:56–68.
32. Rost B. Twilight zone of protein sequence alignments. Protein Eng 1999;12:85–94.
33. Granseth E, Viklund H, Elofsson A. ZPRED: predicting the distance to the membrane center for residues in alpha-helical membrane proteins. Bioinformatics 2006;22:e191–196.
34. Papaloukas C, Granseth E, Viklund H, Elofsson A. Estimating the length of transmembrane helices using Z-coordinate predictions. Protein Sci 2008;17:271–278.
35. Petersen TN, Brunak S, von Heijne G, Nielsen H. SignalP 4.0: discriminating signal peptides from transmembrane regions. Nat Methods 2011;8:785–786.

36. Chen CP, Kernytsky A, Rost B. Transmembrane helix predictions revisited. Protein Sci 2002;11:2774–2791.
37. Efron B, Tibshirani RJ. An introduction to the bootstrap. Chapman & Hall (New York) 1993.
38. Eisenberg D. Three-dimensional structure of membrane and surface proteins. Annu Rev Biochem 1984;53:595–623.
39. Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. Journal of Molecular Biology 1993;232:584–599.
40. Rost B, Liu J, Nair R, Wrzeszczynski KO, Ofran Y. Automatic prediction of protein function. Cellular and Molecular Life Sciences 2003;60:2637–2650.
41. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25:3389–3402.
42. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA Data Mining Software: An Update. SIGKDD Explorations 2009;11:10–18.
43. Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol 1982;157:105–132.
44. Rath EM, Tessier D, Campbell AA, Lee HC, Werner T, Salam NK, Lee LK, Church WB. A benchmark server using high resolution protein structure data, and benchmark results for membrane helix predictions. BMC Bioinformatics 2013;14:111
45. Yachdav G, Kloppmann E, Kajan L, Hecht M, Goldberg T, Hamp T, Honigschmid P, Schafferhans A, Roos M, Bernhofer M, Richter L, Ashkenazy H, Punta M, Schlessinger A, Bromberg Y, Schneider R, Vriend G, Sander C, Ben-Tal N, Rost B. PredictProtein–an open resource for online prediction of protein structural and functional features. Nucleic Acids Res 2014;42:W337–343.

3. PredictProtein - Predicting Protein Structure and Function for 29 Years

3.1. Preface

PredictProtein is one of the oldest web services for the prediction of more than 20 functional and structural features of proteins [113]. It has been running since 1992. Over the years, more and more prediction methods for different features have been added or replaced by improved versions. Thus, the computational resources needed to run PredictProtein and save its results became more and more. In an effort to keep it running, we initiated a project together with the Schneider group at the Luxembourg Centre for Systems Biomedicine (LCSB). Its main goal was to trim old and deprecated code, remove unnecessary and outdated methods, add new ones, and move the infrastructure to a host in Luxembourg.

We sped up and improved the graphical user interface [117–119] and added support for new APIs for programmatic access to the cached prediction results. We added new prediction methods for Gene Ontology (GO) [104], secondary structure [89], and protein-, RNA-, and DNA-binding proteins [120]. Further, we improved our pipeline responsible for searching databases for homolog sequences and alignment generation. In the past, this step often took over an hour to complete. After the update, we were able to reduce it to less than five minutes with the help of the MMseqs2 [74–76] method. Thus, improving the user-experience by significantly reducing the time they spend waiting for results.

As TMSEG [65] (Chapter 2) is also part of the PredictProtein pipeline, it is easy to cross-reference predicted transmembrane proteins (TMP) with features predicted by other methods. For example, GO term and binding predictions might provide further

information about the type of TMP (e.g., transporter, channel, receptor). Thus, making PredictProtein an excellent resource for TMP research.

The PredictProtein web service is freely available at `https://predictprotein.org/`.

Author contribution: Christian Dallago and I contributed substantially to the writing of the manuscript, conceptualization of the new frontend, and implementation of the backend. Tim Karl wrote most of the frontend and backend code, and supported the needed hardware and software. Venkata Satagopam coordinated our efforts at the Luxembourg site. All authors helped with the conceptualization of the new platform and drafted the manuscript.

3.2. Journal Article: Michael Bernhofer *et al.*, Nucleic Acids Research (2021)

Reference: Bernhofer, M., Dallago, C., Karl, T., Satagopam, V., Heinzinger, M., et al. Predictprotein - predicting protein structure and function for 29 years. *Nucleic Acids Res*, 49(W1):W535–W540, 2021. 10.1093/nar/gkab354

Published online 17 May 2021 *Nucleic Acids Research, 2021, Vol. 49, Web Server issue* ***W535–W540***
https://doi.org/10.1093/nar/gkab354

PredictProtein - Predicting Protein Structure and Function for 29 Years

Michael Bernhofer[1,2,†], Christian Dallago[1,2,*,†], Tim Karl[1,†], Venkata Satagopam[3,4,†], Michael Heinzinger[1,2], Maria Littmann[1,2], Tobias Olenyi[1], Jiajun Qiu[1,5], Konstantin Schütze[1], Guy Yachdav[1], Haim Ashkenazy[6,7], Nir Ben-Tal[8], Yana Bromberg[9], Tatyana Goldberg[1], Laszlo Kajan[10], Sean O'Donoghue[11], Chris Sander[12,13,14], Andrea Schafferhans[1,15], Avner Schlessinger[16], Gerrit Vriend[17], Milot Mirdita[18], Piotr Gawron[3], Wei Gu[3,4], Yohan Jarosz[3,4], Christophe Trefois[3,4], Martin Steinegger[19,20], Reinhard Schneider[3,4] and Burkhard Rost[1,21,22,*]

[1]TUM (Technical University of Munich) Department of Informatics, Bioinformatics & Computational Biology - i12, Boltzmannstr 3, 85748 Garching/Munich, Germany, [2]TUM Graduate School CeDoSIA, Boltzmannstr 11, 85748 Garching, Germany, [3]Luxembourg Centre For Systems Biomedicine (LCSB), University of Luxembourg, Campus Belval, House of Biomedicine II, 6 avenue du Swing, L-4367 Belvaux, Luxembourg, [4]ELIXIR Luxembourg (ELIXIR-LU) Node, University of Luxembourg, Campus Belval, House of Biomedicine II, 6 avenue du Swing, L-4367 Belvaux, Luxembourg, [5]Department of Otolaryngology Head & Neck Surgery, The Ninth People's Hospital & Ear Institute, School of Medicine & Shanghai Key Laboratory of Translational Medicine on Ear and Nose Diseases, Shanghai Jiao Tong University, Shanghai, China, [6]Department of Molecular Biology, Max Planck Institute for Developmental Biology, Tübingen, Germany, [7]The Shmunis School of Biomedicine and Cancer Research, George S. Wise Faculty of Life Sciences, Tel Aviv University, 69978 Tel Aviv, Israel, [8]Department of Biochemistry & Molecular Biology, George S. Wise Faculty of Life Sciences, Tel Aviv University, 69978 Tel Aviv, Israel, [9]Department of Biochemistry and Microbiology, Rutgers University, New Brunswick, NJ 08901, USA, [10]Roche Polska Sp. z o.o., Domaniewska 39B, 02–672 Warsaw, Poland, [11]Garvan Institute of Medical Research, Sydney, Australia, [12]Department of Data Sciences, Dana-Farber Cancer Institute, Boston, MA 02215, USA, [13]Department of Cell Biology, Harvard Medical School, Boston, MA 02215, USA, [14]Broad Institute of MIT and Harvard, Boston, MA 02142, USA, [15]HSWT (Hochschule Weihenstephan Triesdorf | University of Applied Sciences), Department of Bioengineering Sciences, Am Hofgarten 10, 85354 Freising, Germany, [16]Department of Pharmacological Sciences, Icahn School of Medicine at Mount Sinai, New York, NY 10029, USA, [17]BIPS, Poblacion Baco, Mindoro, Philippines, [18]Quantitative and Computational Biology, Max Planck Institute for Biophysical Chemistry, Göttingen, Germany, [19]School of Biological Sciences, Seoul National University, Seoul, South Korea, [20]Artificial Intelligence Institute, Seoul National University, Seoul, South Korea, [21]Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany and [22]TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany

Received February 23, 2021; Revised April 06, 2021; Editorial Decision April 21, 2021; Accepted May 10, 2021

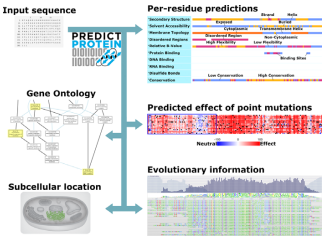
ABSTRACT

Since 1992 *PredictProtein* (https://predictprotein.org) is a one-stop online resource for protein sequence analysis with its main site hosted at the Luxembourg Centre for Systems Biomedicine (LCSB) and queried monthly by over 3,000 users in 2020. *PredictProtein* was the first Internet server for protein predictions. It pioneered combining evolutionary information and machine learning. Given a protein sequence as input, the server outputs multiple sequence alignments, predictions of protein structure in 1D and 2D (secondary structure, solvent accessibility, transmembrane segments, disordered regions, protein flexibility, and disulfide bridges) and predictions of protein function (functional effects of sequence variation or point mutations, Gene Ontology (GO) terms, subcellular localization, and

*To whom correspondence should be addressed. Tel: +49 289 17 811; Email: christian.dallago@tum.de
Correspondence may also be addressed to Burkhard Rost. Email: assistant@rostlab.org
†The authors wish it to be known that, in their opinion, the first four authors should be regarded as joint First Authors.

protein-, RNA-, and DNA binding). *PredictProtein's* infrastructure has moved to the LCSB increasing throughput; the use of MMseqs2 sequence search reduced runtime five-fold (apparently without lowering performance of prediction methods); user interface elements improved usability, and new prediction methods were added. *PredictProtein* recently included predictions from deep learning embeddings (GO and secondary structure) and a method for the prediction of proteins and residues binding DNA, RNA, or other proteins. PredictProtein.org aspires to provide reliable predictions to computational and experimental biologists alike. All scripts and methods are freely available for offline execution in high-throughput settings.

GRAPHICAL ABSTRACT



INTRODUCTION

The sequence is known for far more proteins (1) than experimental annotations of function or structure (2,3). This sequence-annotation gap existed when *PredictProtein* (4,5) started in 1992, and has kept expanding ever since (6). Unannotated sequences contribute crucial evolutionary information to neural networks predicting secondary structure (7,8) that seeded *PredictProtein (PP)* at the European Molecular Biology Laboratory (EMBL) in 1992 (9), the first fully automated, query-driven Internet server providing evolutionary information and structure prediction for any protein. Many other methods predicting aspects of protein function and structure have since joined under the PP roof (4,5,10) now hosted by the Luxembourg Centre of Systems Biomedicine (LCSB).

PP offers an array of structure and function predictions most of which combine machine learning with evolutionary information; now enhanced by a faster alignment algorithm (11,12). A few prediction methods now also use embeddings (13,14) from protein Language Models (LMs) (13–18). Embeddings are much faster to obtain than evolutionary information, yet for many tasks, perform almost as well, or even better (19,20). All PP methods join at PredictProtein.org with interactive visualizations; for some methods, more advanced visualizations are linked (21–23). The reliability of *PredictProtein*, its speed, the continuous integration of well-validated, top methods, and its intuitive interface have attracted thousands of researchers over 29 years of steady operation.

MATERIALS AND METHODS

PredictProtein (PP) provides

multiple sequence alignments (MSAs) and position-specific scoring matrices (PSSMs) computed by a combination of pairwise BLAST (24), PSI-BLAST (25), and MMseqs2 (11,12) on query vs. PDB (26) and query versus UniProt (1). For each residue in the query, the following per-residue predictions are assembled: secondary structure (RePROF/PROFsec (5,27) and ProtBertSec (14)); solvent accessibility (RePROF/PROFacc); transmembrane helices and strands (TMSEG (28) and PROFtmb (29)); protein disorder (Meta-Disorder (30)); backbone flexibility (relative B-values; PROFbval (31)); disulfide bridges (DISULFIND (32)); sequence conservation (ConSurf/ConSeq (33–36)); protein-protein, protein-DNA, and protein-RNA binding residues (ProNA2020 (3)); PROSITE motifs (37); effects of sequence variation (single amino acid variants, SAVs; SNAP2 (38)). For each query per-protein predictions include: transmembrane topology (TMSEG (28)); binary protein-(DNA|RNA|protein) binding (protein binds X or not; ProNA2020 (3)); Gene Ontology (GO) term predictions (goPredSim (19)); subcellular localization (LocTree3 (39)); Pfam (40) domain scans, and some biophysical features. Under the hood, PP computes more results (SOM: PredictProtein Methods; Supplementary Table S1), either as input for frontend methods, or for legacy support.

New: goPredSim embedding-based transfer of Gene Ontology (GO)

goPredSim (19) predicts GO terms by transferring annotations from the most embedding-similar protein. Embeddings are obtained from SeqVec (13); similarity is established through the Euclidean distance between the embedding of a query and a protein with experimental GO annotations. Replicating the conditions of CAFA3 (41) in 2017, goPredSim achieved F_{max} values of 37 ± 2%, 52 ± 2% and 58 ± 2% for BPO (biological process), MFO (molecular function), and CCO (cellular component), respectively (41,42). Using Gene Ontology Annotation (GOA) (43,44) to test 296 proteins annotated after February 2020, goPredSim appeared to reach even slightly higher values that were confirmed by CAFA4 (45).

New: ProtBertSec secondary structure prediction

ProtBertSec predicts secondary structure in three states (helix, strand, other) using ProtBert (14) embeddings derived from training on BFD with almost 3×10^9 proteins (6,46). On a hold-out set from CASP12, ProtBertSec reached a three-state per-residue accuracy of Q3 = 76 ± 1.5% (47). Although below the state-of-the-art (NetSurfP-2.0 (48) at 82%), this method performed on-par with other MSA-based methods, despite itself not using MSAs.

New: ProNA2020 protein–protein, protein–RNA and protein–DNA

ProNA2020 (3) predicts whether or not a protein interacts with other proteins, RNA or DNA (binary), and if so, where

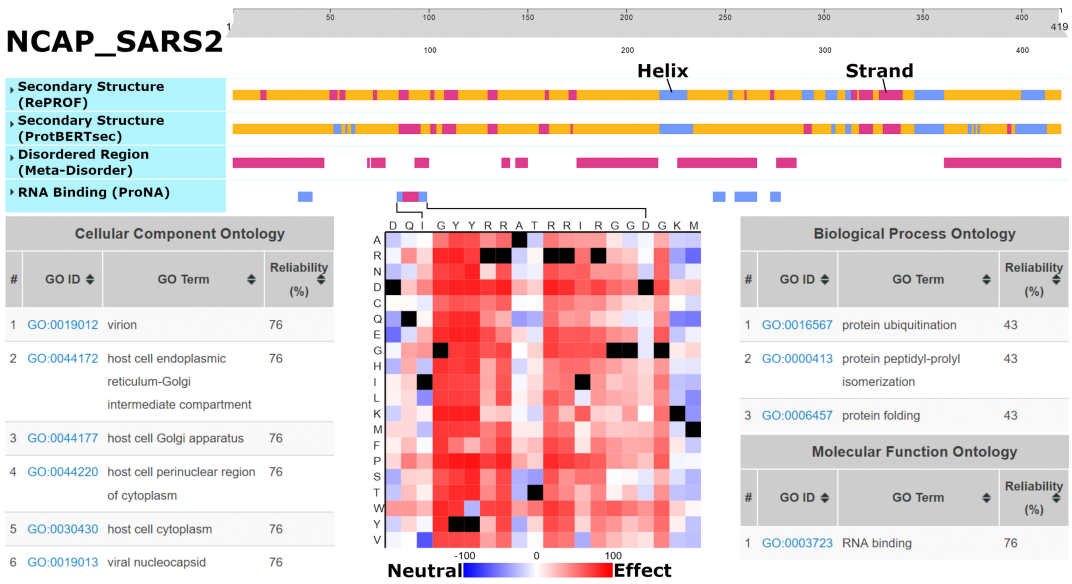


Figure 1. Predictions for SARS-CoV-2 Nucleoprotein (NCAP_SARS2). Underneath the interactive slider at the top: RePROF and ProtBertSec secondary structure (blue: helix; purple: strand; orange: other); Meta-Disorder intrinsically disordered regions (purple); ProNA2020 RNA-binding residues (low confidence: blue; medium confidence: purple). goPredSim transfers of GeneOntology (GO) terms based on embedding similarity (lower left: CCO; lower right: BPO & MFO). SNAP2 predicts the effect of point-mutations on function for the RNA-binding region from I84 to D98 (bottom-center; black: native residue). Link: predictprotein.org/visual_results?req_id=1nAmulUQY$FRPFaP8NTqLW9DzdlTG3B/.

it binds (which residues). The binary per-protein predictions rely on homology and machine learning models employing profile-kernel SVMs (49) and on embeddings from an *in-house* implementation of ProtVec (50). Per-residue predictions (where) use simple neural networks due to data shortage (51–53). ProNA2020 correctly predicted 77 ± 1% of the proteins binding DNA, RNA or protein. In proteins known to bind other proteins, RNA or DNA, ProNA2020 correctly predicted 69 ± 1%, 81 ± 1% and 80 ± 1% of binding residues, respectively.

New: MMseqs2 speedy evolutionary information

Most time-consuming for PP was the search for related proteins in ever growing databases. MMseqs2 (11) finds related sequences blazingly fast and seeds a PSI-BLAST search (25). The query sequence is sent to a dedicated MMseqs2 server that searches for hits against cluster representatives within the UniClust30 (54) and PDB (26) reduced to 70% pairwise percentage sequence identity (PIDE). All hits and their respective cluster members are turned into a MSA and filtered to the 3000 most diverse sequences.

WEB SERVER

Frontend and user interface (UI)

Users query PredictProtein.org by submitting a protein sequence. Results are available in seconds for sequences that had been submitted recently (cf. *PPcache* next section), or within up to 30 min if predictions are recomputed. Per-residue predictions are displayed online via ProtVista (55), which allows to zoom into any sequential protein region (Supplementary Figure S1), and are grouped by category (e.g. secondary structure), which can be expanded to display more detail (e.g. helix, strand, other). On the results page, links to visualize MSAs through *AlignmentViewer* (56) are available. More predictions can be accessed through a menu on the left, e.g. *Gene Ontology Terms*, *Effect of Point Mutations* and *Subcellular Localization*. Prediction views include references and details of outputs, as well as rich visualizations, e.g. GO trees for GO predictions and cell images with highlighted predicted locations for subcellular localization predictions (57).

PPcache, backend and programmatic access

The PP backend lives at LCSB, allowing for up to 48 parallel queries. Results are stored on disc in the *PPcache* (5). Users submitting sequences for which results were over the last two years obtain results immediately. Using the bio-embeddings pipeline (58), the PPcache is enriched by embeddings and embedding-based predictions on the fly. For all methods displayed on the frontend, JSON files compliant with *ProtVista* (55) are available via REST APIs (SOM: Programmatic access), and are in use by external services such as the protein 3D structure visualization suite *Aquaria* (21,23) and by *MolArt* (22).

PredictProtein is available for local use

All results displayed on and downloadable from PP are available through Docker (59) and as source code for local and cloud execution (available at github.com/rostlab).

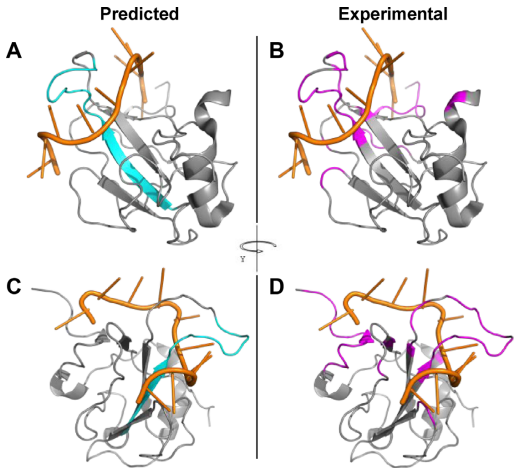


Figure 2. Experimental and predicted RNA-binding residues for NCAP2_SARS2. Predicted (via ProNA2020, in cyan, panels **A** and **C**) and observed (within 5Å, in magenta, panels **B** and **D**) RNA-binding residues for the SARS-CoV-2 nucleoprotein (gray) complexed with a 10-mer ssRNA (orange), PDB structure 7ACT (61). Two-third of the predictions are correct (precision = 0.73, recall = 0.20), which is around the expected average performance reported by ProNA2020. The important sequence consecutive central strand and loop are predicted well, while several short sequence segments that are far away in sequence space but close in structure space are missed, which is expected as ProNA2020 has no notion of 3D structure, i.e., cannot identify 'binding sites'. Panels A and B show a different orientation than panels C and D.

USE CASE

We demonstrate PredictProtein.org tools through predictions of the novel coronavirus SARS-CoV-2 (NCBI:txid2697049) nucleoprotein (UniProt identifier P0DTC9/NCAP_SARS2; Figure 1; SOM: Use Case; Supplementary Figure S2). NCAP_SARS2 has 419 residues and interacts with itself (experimentally verified). Sequence similarity and automatic assignment via UniRule (60) suggest NCAP is RNA-binding (binding with the viral genome), binding with the membrane protein M (UniProt identifier P0DTC5/VME1_SARS2), and is fundamental for virion assembly. goPredSim (19) transferred GO terms from other proteins for MFO (*RNA-binding*; GO:0003723; ECO:0000213) and CCO (compartments in the host cell and viral nucleocapsid; GO:0019013; GO:0044172; GO:0044177; GO:0044220; GO:0030430; ECO:0000255) matching annotations found in UniProt (1). While it missed the experimentally verified MFO term *identical protein binding* (GO:0042802), goPredSim predicted *protein folding* (GO:0006457) and *protein ubiquitination* (GO:0016567) suggesting the nucleoprotein to be involved in biological processes requiring protein binding. ProNA2020 (3) predicts RNA-binding regions, the one with highest confidence between I84 (Isoleucine at position 84) and D98 (Aspartic Acid at 98) (protein sequence in SOM: Use Case). While high resolution experimental data on binding is not available, an NMR structure of the SARS-CoV-2 nucleocapsid phosphoprotein N-terminal domain in complex with 10mer ssRNA (PDB identifier 7ACT (61)) supports the predicted RNA-binding site (Figure 2). Additionally, SNAP2 (38) predicts single amino acid variants (SAVs) in that region to likely affect function, reinforcing the hypothesis that this is a functionally relevant site. Although using different input information (evolutionary vs. embeddings), RePROF (5) and ProtBertSec (14) both predict an unusual content exceeding 70% non-regular (neither helix nor strand) secondary structure, suggesting that most of the nucleoprotein might not form regular structure. This is supported by Meta-Disorder (30) predicting 53% overall disorder. Secondary structure predictions match well high-resolution experimental structures of the nucleoprotein not in complex (e.g., PDB 6VYO (62); 6WJI (63)). Both secondary structure prediction methods managed to zoom into the ordered regions of the protein and predicted e.g., the five beta-strands that are formed within the sequence range I84 (Isoleucine) to A134 (Alanine), and the two helices formed within the sequence range spanned from F346 (Phenylalanine) to T362 (Tyrosine).

CONCLUSION

For almost three decades (preceding Google) *PredictProtein* (PP) has been offering predictions for proteins. PP is the oldest prediction Internet server, online for 6-times as long as most other servers (64–66). It pioneered combining machine learning with evolutionary information and making predictions freely accessible online. While the sequence-annotation gap continues to grow, the sequence-structure gap might be bridged in the near future (67). For the time being, servers such as PP, providing a one-stop solution to predictions from many sustained, novel tools are needed. Now, PP is the first server to offer fast embedding-based predictions of structure (ProtBertSec) and function (goPredSim). By slashing runtime for PSSMs from 72 to 4 min through MMseqs2 and better LCSB hardware, PP also delivers evolutionary information-based predictions fast! Instantaneously if the query is in the precomputed *PPcache*. For heavy use, the offline Docker containers provide predictors out-of-the-box. At no cost to users, *PredictProtein* offers to quickly shine light on proteins for which little is known using well validated prediction methods.

DATA AVAILABILITY

Freely accessible webserver PredictProtein.org; Source and docker images: github.com/rostlab.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

Maintaining *PredictProtein* over three decades has been tough; many colleagues have helped with hands and brains, developers, and users alike. Thanks to all of you! Please find most contributors in Supplementary Table S2 or at predictprotein.org/credits. In particular, thanks to Noua Toukourou and Maharshi Vyas (both LCSB) for invaluable

help with hardware and software; to David Hoksza (Charles U, Prague) for his work on MolArt; to Marco Punta (IRCCS Milano) for his long-term support; to Inga Weise (TUM) for support with many aspects; to Roy Omond (Blue Bubble, Cambridge), Antoine de Daruvar (Univ. Bordeaux), Yanay Ofran (Bar-Ilan Univ.), Jinfeng Liu (Genentech), Tobias Hamp, Maximilian Hecht, Edda Kloppmann (all previously TUM) for contributing methods and code in the past; Johannes Söding for providing resources to develop and maintain MMseqs2.

FUNDING

Michael Bernhofer was supported by the Competence Network for Scientific High Performance Computing in Bavaria [KONWIHR-III BG.DAF]; Christian Dallago is supported by the Deutsche Forschungsgemeinschaft (DFG) [RO 1320/4-1]; Bundesministerium für Bildung und Forschung (BMBF) [031L0168]; Software Campus 2.0 (TU München), BMBF [01IS17049]; Milot Mirdita acknowledges support from the ERC's Horizon 2020 Framework Programme [`Virus-X', project no. 685778]; BMBF CompLifeSci project horizontal4meta. Martin Steinegger acknowledges support from the National Research Foundation of Korea grant funded by the Korean government (MEST) [2019R1A6A1A10073437, NRF-2020M3A9G7103933]; Creative-Pioneering Researchers Program through Seoul National University; Nir Ben-Tal acknowledges the support of Israeli Science Foundation (ISF) [450/16]; Abraham E. Kazan Chair in Structural Biology, Tel Aviv University; Haim Ashkenazy was supported by Humboldt Research Fellowship for Postdoctoral Researchers of the Alexander von Humboldt Foundation; The PredictProtein web server is hosted by ELIXIR-LU, the Luxembourgish node of the European life-science infrastructure. Funding for open access charge: Library of the Technical University of Munich.
Conflict of interest statement. None declared.

REFERENCES

1. The UniProt Consortium (2021) UniProt: the universal protein knowledgebase in 2021. *Nucleic. Acids Res.*, **49**, D480–D489.
2. Rost,B., Liu,J., Nair,R., Wrzeszczynski,K.O. and Ofran,Y. (2003) Automatic prediction of protein function. *Cell. Mol. Life Sci.*, **60**, 2637–2650.
3. Qiu,J., Bernhofer,M., Heinzinger,M., Kemper,S., Norambuena,T., Melo,F. and Rost,B. (2020) ProNA2020 predicts protein–DNA, protein–RNA, and protein–protein binding proteins and residues from sequence. *J. Mol. Biol.*, **432**, 2428–2443.
4. Rost,B. (1996) PHD: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol.*, **266**, 525–539.
5. Yachdav,G., Kloppmann,E., Kajan,L., Hecht,M., Goldberg,T., Hamp,T., Hönigschmid,P., Schafferhans,A., Roos,M., Bernhofer,M. *et al.* (2014) PredictProtein—an open resource for online prediction of protein structural and functional features. *Nucleic Acids Res.*, **42**, W337–W343.
6. Steinegger,M. and Söding,J. (2018) Clustering huge protein sequence sets in linear time. *Nat. Commun.*, **9**, 2542.
7. Rost,B. and Sander,C. (1993) Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Natl. Acad. Sci. U.S.A.*, **90**, 7558–7562.
8. Rost,B. and Sander,C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, **232**, 584–599.
9. Rost,B. and Sander,C. (1992) Jury returns on structure prediction. *Nature*, **360**, 540.
10. Kajan,L., Yachdav,G., Vicedo,E., Steinegger,M., Mirdita,M., Angermuller,C., Bohm,A., Domke,S., Ertl,J., Mertes,C. *et al.* (2013) Cloud prediction of protein structure and function with PredictProtein for Debian. *Biomed. Res. Int.*, **2013**, 398968.
11. Steinegger,M. and Söding,J. (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, **35**, 1026–1028.
12. Mirdita,M., Steinegger,M. and Söding,J. (2019) MMseqs2 desktop and local web server app for fast, interactive sequence searches. *Bioinformatics*, **35**, 2856–2858.
13. Heinzinger,M., Elnaggar,A., Wang,Y., Dallago,C., Nechaev,D., Matthes,F. and Rost,B. (2019) Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, **20**, 723.
14. Elnaggar,A., Heinzinger,M., Dallago,C., Rihawi,G., Wang,Y., Jones,L., Gibbs,T., Feher,T., Angerer,C., Bhowmik,D. *et al.* (2020) ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. arXiv doi: https://arxiv.org/abs/2007.06225, 04 May 2021, preprint: not peer reviewed.
15. Alley,E.C., Khimulya,G., Biswas,S., AlQuraishi,M. and Church,G.M. (2019) Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, **16**, 1315–1322.
16. AlQuraishi,M. (2019) End-to-end differentiable learning of protein structure. *Cell Syst.*, **8**, 292–301.
17. Rao,R., Bhattacharya,N., Thomas,N., Duan,Y., Chen,P., Canny,J., Abbeel,P. and Song,Y. (2019) Evaluating Protein Transfer Learning with TAPE. In: Wallach,H., Larochelle,H., Beygelzimer,A., d'Alché-Buc,F., Fox,E. and Garnett,R. (eds). *Advances in Neural Information Processing Systems*. Vol. **32**. Curran Associates, Inc., pp. 9689–9701.
18. Rives,A., Meier,J., Sercu,T., Goyal,S., Guo,D., Lin,Z., Liu,J., Guo,D., Ott,M., Zitnick,C.L., Ma,J. and Fergus,R. (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA*, **118**, e2016239118.
19. Littmann,M., Heinzinger,M., Dallago,C., Olenyi,T. and Rost,B. (2021) Embeddings from deep learning transfer GO annotations beyond homology. *Sci. Rep.*, **11**, 1160.
20. Rao,R., Ovchinnikov,S., Meier,J., Rives,A. and Sercu,T. (2020) Transformer protein language models are unsupervised structure learners. bioRxiv doi: https://doi.org/10.1101/2020.12.15.422761, 15 December 2020, preprint: not peer reviewed.
21. O'Donoghue,S.I., Sabir,K.S., Kalemanov,M., Stolte,C., Wellmann,B., Ho,V., Roos,M., Perdigão,N., Buske,F.A., Heinrich,J. *et al.* (2015) Aquaria: simplifying discovery and insight from protein structures. *Nat. Methods*, **12**, 98–99.
22. Hoksza,D., Gawron,P., Ostaszewski,M. and Schneider,R. (2018) MolArt: a molecular structure annotation and visualization tool. *Bioinformatics*, **34**, 4127–4128.
23. O'Donoghue,S.I., Schafferhans,A., Sikta,N., Stolte,C., Kaur,S., Ho,B.K., Anderson,S., Procter,J., Dallago,C., Bordin,N. *et al.* (2020) SARS-CoV-2 structural coverage map reveals state changes that disrupt host immunity bioinformatics. bioRxiv doi: https://doi.org/10.1101/2020.07.16.207308, 28 September 2020, preprint: not peer reviewed.
24. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
25. Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
26. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
27. Rost,B. (2001) Protein secondary structure prediction continues to rise. *J. Struct. Biol.*, **134**, 204–218.
28. Bernhofer,M., Kloppmann,E., Reeb,J. and Rost,B. (2016) TMSEG: novel prediction of transmembrane helices. *Proteins*, **84**, 1706–1716.
29. Bigelow,H. and Rost,B. (2006) PROFtmb: a web server for predicting bacterial transmembrane beta barrel proteins. *Nucleic Acids. Res.*, **34**, W186–W188.

30. Schlessinger,A., Punta,M., Yachdav,G., Kajan,L. and Rost,B. (2009) Improved disorder prediction by combination of orthogonal approaches. *PLoS One*, **4**, e4433.
31. Schlessinger,A., Yachdav,G. and Rost,B. (2006) PROFbval: predict flexible and rigid residues in proteins. *Bioinforma. Oxf. Engl.*, **22**, 891–893.
32. Ceroni,A., Passerini,A., Vullo,A. and Frasconi,P. (2006) DISULFIND: a disulfide bonding state and cysteine connectivity prediction server. *Nucleic Acids Res.*, **34**, W177–W181.
33. Berezin,C., Glaser,F., Rosenberg,J., Paz,I., Pupko,T., Fariselli,P., Casadio,R. and Ben-Tal,N. (2004) ConSeq: the identification of functionally and structurally important residues in protein sequences. *Bioinforma. Oxf. Engl.*, **20**, 1322–1324.
34. Ashkenazy,H., Erez,E., Martz,E., Pupko,T. and Ben-Tal,N. (2010) ConSurf 2010: calculating evolutionary conservation in sequence and structure of proteins and nucleic acids. *Nucleic Acids Res.*, **38**, W529–W533.
35. Celniker,G., Nimrod,G., Ashkenazy,H., Glaser,F., Martz,E., Mayrose,I., Pupko,T. and Ben-Tal,N. (2013) ConSurf: using evolutionary data to raise testable hypotheses about protein function. *Isr. J. Chem.*, **53**, 199–206.
36. Ashkenazy,H., Abadi,S., Martz,E., Chay,O., Mayrose,I., Pupko,T. and Ben-Tal,N. (2016) ConSurf 2016: an improved methodology to estimate and visualize evolutionary conservation in macromolecules. *Nucleic Acids Res.*, **44**, W344–W350.
37. Sigrist,C.J.A., de Castro,E., Cerutti,L., Cuche,B.A., Hulo,N., Bridge,A., Bougueleret,L. and Xenarios,I. (2013) New and continuing developments at PROSITE. *Nucleic Acids Res.*, **41**, D344–347.
38. Hecht,M., Bromberg,Y. and Rost,B. (2015) Better prediction of functional effects for sequence variants. *BMC Genomics*, **16 (Suppl 8)**, S1.
39. Goldberg,T., Hecht,M., Hamp,T., Karl,T., Yachdav,G., Ahmed,N., Altermann,U., Angerer,P., Ansorge,S., Balasz,K. *et al.* (2014) LocTree3 prediction of localization. *Nucleic Acids Res.*, **42**, W350–W355.
40. El-Gebali,S., Mistry,J., Bateman,A., Eddy,S.R., Luciani,A., Potter,S.C., Qureshi,M., Richardson,L.J., Salazar,G.A., Smart,A. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.
41. Zhou,N., Jiang,Y., Bergquist,T.R., Lee,A.J., Kacsoh,B.Z., Crocker,A.W., Lewis,K.A., Georghiou,G., Nguyen,H.N., Hamid,M.N. *et al.* (2019) The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol.*, **20**, 244.
42. Jiang,Y., Oron,T.R., Clark,W.T., Bankapur,A.R., D'Andrea,D., Lepore,R., Funk,C.S., Kahanda,I., Verspoor,K.M., Ben-Hur,A. *et al.* (2016) An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.*, **17**, 184.
43. Camon,E., Magrane,M., Barrell,D., Lee,V., Dimmer,E., Maslen,J., Binns,D., Harte,N., Lopez,R. and Apweiler,R. (2004) The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res.*, **32**, D262–D266.
44. Huntley,R.P., Sawford,T., Mutowo-Meullenet,P., Shypitsyna,A., Bonilla,C., Martin,M.J. and O'Donovan,C. (2015) The GOA database: gene ontology annotation updates for 2015. *Nucleic Acids Res.*, **43**, D1057–D1063.
45. El-Mabrouk,N. and Slonim,D.K. (2020) ISMB 2020 proceedings. *Bioinformatics*, **36**, i1–i2.
46. Steinegger,M., Mirdita,M. and Söding,J. (2019) Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods*, **16**, 603–606.
47. Abriata,L.A., Tamò,G.E., Monastyrskyy,B., Kryshtafovych,A. and Peraro,M.D. (2018) Assessment of hard target modeling in CASP12 reveals an emerging role of alignment-based contact prediction methods. *Proteins Struct. Funct. Bioinforma.*, **86**, 97–112.
48. Klausen,M.S., Jespersen,M.C., Nielsen,H., Jensen,K.K., Jurtz,V.I., Sønderby,C.K., Sommer,M.O.A., Winther,O., Nielsen,M., Petersen,B. *et al.* (2019) NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. *Proteins Struct. Funct. Bioinforma.*, **87**, 520–527.
49. Hamp,T., Goldberg,T. and Rost,B. (2013) Accelerating the original profile kernel. *PLoS One*, **8**, e68459.
50. Asgari,E., McHardy,A.C. and Mofrad,M.R.K. (2019) Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). *Sci. Rep.*, **9**, 3577.
51. Norambuena,T. and Melo,F. (2010) The protein-DNA interface database. *BMC Bioinformatics*, **11**, 262.
52. Lewis,B.A., Walia,R.R., Terribilini,M., Ferguson,J., Zheng,C., Honavar,V. and Dobbs,D. (2011) PRIDB: a protein-RNA interface database. *Nucleic. Acids. Res.*, **39**, D277–D282.
53. Hamp,T. and Rost,B. (2015) Evolutionary profiles improve protein-protein interaction prediction from sequence. *Bioinforma. Oxf. Engl.*, **31**, 1945–1950.
54. Mirdita,M., von den Driesch,L., Galiez,C., Martin,M.J., Söding,J. and Steinegger,M. (2017) Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic. Acids. Res.*, **45**, D170–D176.
55. Watkins,X., Garcia,L.J., Pundir,S., Martin,M.J. and Consortium,U. (2017) ProtVista: visualization of protein sequence annotations. *Bioinformatics*, **33**, 2040–2041.
56. Reguant,R., Antipin,Y., Sheridan,R., Dallago,C., Diamantoukos,D., Luna,A., Sander,C. and Gauthier,N.P. (2020) AlignmentViewer: sequence analysis of large protein families. *F1000Research*, **9**, 213.
57. Dallago,C., Goldberg,T., Andrade-Navarro,M.A., Alanis-Lobato,G. and Rost,B. (2020) Visualizing human protein-protein interactions and subcellular localizations on cell images through CellMap. *Curr. Protoc. Bioinforma.*, **69**, e97.
58. Dallago,C., Schütze,K., Heinzinger,M., Olenyi,T., Littmann,M., Lu,A.X., Yang,K.K., Min,S., Yoon,S., Morton,J.T. *et al.* (2021) Learned embeddings from deep learning to visualize and predict protein sets. *Curr. Protoc. Bioinforma.*, **1**, e113.
59. Merkel,D. (2014) Docker: lightweight linux containers for consistent development and deployment. *Linux J.*, **2014**, 2.
60. MacDougall,A., Volynkin,V., Saidi,R., Poggioli,D., Zellner,H., Hatton-Ellis,E., Joshi,V., O'Donovan,C., Orchard,S., Auchincloss,A.H. *et al.* (2020) UniRule: a unified rule resource for automatic annotation in the UniProt Knowledgebase. *Bioinformatics*, **36**, 4643–4648.
61. Dinesh,D.C., Chalupska,D., Silhan,J., Koutna,E., Nencka,R., Veverka,V. and Boura,E. (2020) Structural basis of RNA recognition by the SARS-CoV-2 nucleocapsid phosphoprotein. *PLoS Pathog.*, **16**, e1009100.
62. Chang,C., Michalska,K., Jedrzejczak,R., Maltseva,N., Endres,M., Godzik,A., Kim,Y. and Joachimiak,A. (2020) Crystal structure of RNA binding domain of nucleocapsid phosphoprotein from SARS coronavirus 2. doi:10.2210/pdb6vyo/pdb.
63. Minasov,G., Shuvalova,L., Wiersum,G. and Satchell,K. (2020) 2.05 angstrom resolution crystal structure of C-terminal dimerization domain of nucleocapsid phosphoprotein from SARS-CoV-2. doi:10.2210/pdb6wji/pdb.
64. Schultheiss,S.J., Münch,M.-C., Andreeva,G.D. and Rätsch,G. (2011) Persistence and availability of Web services in computational biology. *PLoS One*, **6**, e24914.
65. Wren,J.D., Georgescu,C., Giles,C.B. and Hennessey,J. (2017) Use it or lose it: citations predict the continued online availability of published bioinformatics resources. *Nucleic Acids. Res.*, **45**, 3627–3633.
66. Kern,F., Fehlmann,T. and Keller,A. (2020) On the lifetime of bioinformatics web services. *Nucleic Acids Res.*, **48**, 12523–12533.
67. Callaway,E. (2020) 'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures. *Nature*, **588**, 203–204.

4. TMbed: Transmembrane Proteins Predicted through Language Model Embeddings

4.1. Preface

Adapting the newest breakthroughs in natural language processing (NLP) to computation biology gave rise to so-called protein language models (pLM) [89, 91–93]. Trained on millions or billions of protein sequences, those pLMs are able to learn how naturally occurring sequences are structured. Using this condensed knowledge of whole protein databases, they are able to generate information-rich vector-representations (embeddings) for each residue in a protein sequence. Those embedding are then used as input features to downstream models, which often perform on par with methods using evolutionary information [89, 104–106]. However, one advantage of pLMs is that it takes usually less time to generate embeddings than it takes to search a database and generate a multiple sequence alignment. Further, pLMs can generate embeddings even for protein sequence that have only few or no known homologs.

For our second method, TMbed, we decided to replace the evolutionary information with embeddings generated by the pLM named ProtT5-XL-U50 [89]. This pLM was pre-trained on billions of protein sequences from BFD [75, 103] and later fine-tuned on the non-redundant UniRef50 [102]. Those embeddings are the sole input to our downstream model, which consists of a small convolutional neural network (CNN), a Gaussian filter to smooth the predicted class scores, and a Viterbi decoder to generate the most likely output topology. In total, TMbed predicts one of five different states for each residue: transmembrane helix (TMH), transmembrane beta strand (TMB), signal

peptide, inside, and outside. This makes TMbed one of the few prediction methods to predict both types of transmembrane proteins (TMP): alpha helical and beta-barrel.

Our training data consisted of TMPs from the Orientations of Proteins in Membranes [5] (OPM) database and soluble proteins with and without signal peptides from the SignalP6.0 [110] data set. After removing redundant sequences, we were left with a quite unbalanced data set: 5859 soluble proteins, 593 alpha-helical TMPs, and 65 beta-barrel TMPs. Surprisingly however, this 100:10:1 split did not negatively affect the prediction performance.

Evaluating the prediction performance of TMbed using a nested cross-validation, we showed that it was able to compete with the current state-of-the-art methods for both alpha-helical and beta-barrel TMPs. It successfully recognized $98 \pm 1\%$ and $94 \pm 8\%$ of all alpha-helical and beta-barrel TMPs in the data set, respectively, and misclassified less than 1% of all soluble proteins. On a per-segment level, TMbed correctly placed $88 \pm 1\%$ of all TMHs and $95 \pm 4\%$ of all TMBs within five residues of their annotated location. Even though it was not the primary focus of our method, it was even comparable to SignalP6.0 in terms of signal peptide prediction (TMbed detection rates of $99 \pm 1\%$ per-protein and $93 \pm 1\%$ per-segment). During a closer comparison of TMbed with DeepTMHMM [68] we re-discovered how much annotations from databases can differ [65, 121]. Although both method were developed at about the same time, the length distributions of the TMH and TMB segments were quite different, in both the training data and the predictions. This highlighted two important facts: 1) both models were quite adept at learning the underlying distributions, and 2) there seems to be a distinct lack of a "gold standard" for TMH and TMB annotations. For example, the data set of DeepTMHMM contained an unusual amount of TMHs with 21 residues. This is most likely related to the fact that many of the automated and later curated annotations in UniProtKB do have exactly 21 residues, which in turn might date back to methods like TOP-PRED [40] and TMHMM [47, 48] that strongly favored this length.

TMbed is freely available on GitHub (`https://github.com/BernhoferM/TMbed`) and as part of the LambdaPP [122] (`https://embed.predictprotein.org/`) web service.

Author contribution: I designed and developed the TMbed method, collected all data sets, and performed all evaluations. All authors drafted the manuscript.

4.2. Journal Article: Michael Bernhofer *et al.*, BMC Bioinformatics (2022)

Reference: Bernhofer, M. and Rost, B. Tmbed: transmembrane proteins predicted through language model embeddings. *BMC Bioinformatics*, 23(1):326, 2022. 10.1186/s12859-022-04873-x

Bernhofer and Rost *BMC Bioinformatics* (2022) 23:326
https://doi.org/10.1186/s12859-022-04873-x

BMC Bioinformatics

RESEARCH Open Access

TMbed: transmembrane proteins predicted through language model embeddings

Michael Bernhofer[1,2*] and Burkhard Rost[1,3,4]

*Correspondence:
bernhoferm@rostlab.org

[1] Department of Informatics, Bioinformatics and Computational Biology - i12, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany
[2] TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Boltzmannstr. 11, 85748 Garching, Germany
[3] Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching, Germany
[4] TUM School of Life Sciences Weihenstephan (TUM-WZW), Alte Akademie 8, Freising, Germany

Abstract

Background: Despite the immense importance of transmembrane proteins (TMP) for molecular biology and medicine, experimental 3D structures for TMPs remain about 4–5 times underrepresented compared to non-TMPs. Today's top methods such as AlphaFold2 accurately predict 3D structures for many TMPs, but annotating transmembrane regions remains a limiting step for proteome-wide predictions.

Results: Here, we present TMbed, a novel method inputting embeddings from protein Language Models (pLMs, here ProtT5), to predict for each residue one of four classes: transmembrane helix (TMH), transmembrane strand (TMB), signal peptide, or other. TMbed completes predictions for entire proteomes within hours on a single consumer-grade desktop machine at performance levels similar or better than methods, which are using evolutionary information from multiple sequence alignments (MSAs) of protein families. On the per-protein level, TMbed correctly identified 94 ± 8% of the beta barrel TMPs (53 of 57) and 98 ± 1% of the alpha helical TMPs (557 of 571) in a non-redundant data set, at false positive rates well below 1% (erred on 30 of 5654 non-membrane proteins). On the per-segment level, TMbed correctly placed, on average, 9 of 10 transmembrane segments within five residues of the experimental observation. Our method can handle sequences of up to 4200 residues on standard graphics cards used in desktop PCs (e.g., NVIDIA GeForce RTX 3060).

Conclusions: Based on embeddings from pLMs and two novel filters (Gaussian and Viterbi), TMbed predicts alpha helical and beta barrel TMPs at least as accurately as any other method but at lower false positive rates. Given the few false positives and its outstanding speed, TMbed might be ideal to sieve through millions of 3D structures soon to be predicted, e.g., by AlphaFold2.

Keywords: Protein language models, Protein structure prediction, Transmembrane protein prediction

Background

Structural knowledge of TMPs 4–5 fold underrepresented

Transmembrane proteins (TMP) account for 20–30% of all proteins within any organism [1, 2]; most TMPs cross the membrane with transmembrane helices (TMH). TMPs crossing with transmembrane beta strands (TMB), forming beta barrels, have been estimated to account for 1–2% of all proteins in Gram-negative bacteria; this variety is also

present in mitochondria and chloroplasts [3]. Membrane proteins facilitate many essential processes, including regulation, signaling, and transportation, rendering them targets for most known drugs [4, 5]. Despite this immense relevance for molecular biology and medicine, only about 5% of all three-dimensional (3D) structures in the PDB [6, 7] constitute TMPs [8–10].

Accurate 3D predictions available for proteomes need classification

The prediction of protein structure from sequence leaped in quality through AlphaFold2 [11], Nature's method of the year 2021 [12]. Although AlphaFold2 appears to provide accurate predictions for only very few novel "folds", it importantly increases the width of structural coverage [13]. AlphaFold2 seems to work well on TMPs [14], but for proteome-wide high-throughput studies, we still need to filter out membrane proteins from the structure predictions. Most state-of-the-art (SOTA) TMP prediction methods rely on evolutionary information in the form of multiple sequence alignments (MSA) to achieve their top performance. In our tests we included 13 such methods, namely BetAware-Deep [15], BOCTOPUS2 [16], CCTOP [17, 18], HMM-TM [19–21], OCTOPUS [22], Philius [23], PolyPhobius [24], PRED-TMBB2 [20, 21, 25], PROFtmb [3], SCAMPI2 [26], SPOCTOPUS [27], TMSEG [28], and TOPCONS2 [29].

pLMs capture crucial information without MSAs

Mimicking recent advances of Language Models (LM) in natural language processing (NLP), protein Language Models (pLMs) learn to reconstruct masked parts of protein sequences based on the unmasked local and global information [30–37]. Such pLMs, trained on billions of protein sequences, implicitly extract important information about protein structure and function, essentially capturing aspects of the "language of life" [32]. These aspects can be extracted from the last layers of the deep learning networks into vectors, referred to as embeddings, and used as exclusive input to subsequent methods trained in supervised fashion to successfully predict aspects of protein structure and function [30–34, 36, 38–43]. Often pLM-based methods outperform SOTA methods, which are using evolutionary information on top, and they usually require substantially fewer compute resources. Just before submitting this work, we became aware of another pLM-based TM-prediction method, namely DeepTMHMM [44] using ESM-1b [36] embeddings, and included it in our comparisons.

Here, we combined embeddings generated by the ProtT5 [34] pLM with a simple convolutional neural network (CNN) to create a fast and highly accurate prediction method for alpha helical and beta barrel transmembrane proteins and their overall inside/outside topology. Our new method, TMbed, predicted the presence and location of any TMBs, TMHs, and signal peptides for all proteins of the human proteome within 46 min on our server machine (Additional file 1: Table S1) at the same or better level of performance as other methods, which require substantially more time.

Materials and methods

Data set: membrane proteins (TMPs)

We collected all primary structure files for alpha helical and beta barrel transmembrane proteins (TMP) from OPM [45] and mapped their PDB [6, 7] chain identifiers (PDB-id)

to UniProtKB [46] through SIFTS [47, 48]. Toward this end, we discarded all chimeric chains, all models, and all chains for which OPM failed to map any transmembrane start or end position. This resulted in 2,053 and 206 sequence-unique PDB chains for alpha helical and beta barrel TMPs, respectively.

We used the ATOM coordinates inside the OPM files to assign the inside/outside orientation of sequence segments not within the membrane. We manually inspected inconsistent annotations (e.g., if both ends of a transmembrane segment had the same inside/outside orientation) and cross-referenced them with PDBTM [49–51], PDB, and UniProtKB. We then either corrected such inconsistent annotations or discarded the whole sequence. As OPM does not include signal peptide annotations, we compared our TMP data sets to the set used by SignalP 6.0 [52] and all sequences in UniProtKB/Swiss-Prot with experimentally annotated signal peptides using CD-HIT [53, 54]. For any matches with at least 95% global sequence identity (PIDE), we transferred the signal peptide annotation onto our TMPs. We removed all sequences with fewer than 50 residues to avoid noise from incorrect sequencing fragments, and all sequences with over 15,000 residues to save energy (lower computational costs).

Finally, we removed redundant sequences from the two TMP data sets by clustering them with MMseqs2 [55] to at most 20% local pairwise sequence identity (PIDE) with 40% minimum alignment coverage, i.e., no pair had more than 20% PIDE for any local alignment covering at least 40% of the shorter sequence. The final non-redundant TMP data sets contained 593 alpha helical TMPs and 65 beta barrel TMPs, respectively.

Data set: globular non-membrane proteins

We used the SignalP 6.0 (SP6) dataset for our globular proteins. As the SP6 dataset contained only the first 70 residues of each protein, we took the full sequences from UniProtKB/Swiss-Prot and transferred the signal peptide annotations. To remove any potential membrane proteins from this non-TMP data set, we compared it with CD-HIT [53, 54] against three other data sets: (1) our TMP data sets before redundancy reduction, (2) all protein sequences from UniProtKB/Swiss-Prot with any annotations of transmembrane segments, and (3) all proteins from UniProtKB/Swiss-Prot with any subcellular location annotations for membrane. We removed all proteins from our non-TMP data set with more than 60% global PIDE to any protein in sets 1–3. Again, we dropped all sequences with less than 50 or more than 15,000 residues and applied the same redundancy reduction as before (20% PIDE at 40% alignment coverage). The final non-redundant data set contained 5,859 globular, water-soluble non-TMP proteins; 698 of these have a signal peptide.

Additional redundancy reduction

One anonymous reviewer spotted homologs in our data set after the application of the above protocol. To address this problem, we performed another iteration of redundancy reduction for each of the three data sets using CD-HIT at 20% PIDE. In order to save energy (i.e., avoid retraining our model), we decided to remove clashes for the evaluation, i.e., if two proteins shared more than 20% PIDE, we removed both from the data set (as TMbed was trained on both in the cross-validation protocol). Thereby, this second iteration removed 235 proteins: 8 beta barrel TMPs, 22 alpha helical TMPs, and 205

globular, non-membrane proteins. Our final test data sets included 57 beta barrel TMPs, 571 alpha helical TMPs, and 5654 globular, non-membrane proteins.

Membrane re-entrant regions

Besides transmembrane segments that cross the entire membrane, there are also others, namely membrane segments that briefly enter and exit the membrane on the same side. These are referred to as re-entrant regions [56, 57]. Although rare, some methods explicitly predict them [17, 18, 22, 27, 58]. However, as OPM does not explicitly annotate such regions and since our data set already had a substantial class imbalance between beta barrel TMPs, alpha helical TMPs and, globular proteins, we decided not to predict re-entrant regions.

Embeddings

We generated embeddings with protein Language Models (pLMs) for our data sets using a transformer-based pLM ProtT5-XL-U50 (short: ProtT5) [34]. We discarded the decoder part of ProtT5, keeping only the encoder for increased efficiency (note: encoder embeddings are more informative [34]). The encoder model converts a protein sequence into an embedding matrix that represents each residue in the protein, i.e., each position in the sequence, by a 1024-dimensional vector containing global and local contextualized information. We converted the ProtT5 encoder from 32-bit to 16-bit floating-point format to reduce the memory footprint on the GPU. We took the pre-trained ProtT5 model as is without any further task-specific fine-tuning.

We chose ProtT5 over other embedding models, such as ESM-1b [36], based on our experience with the model and comparisons during previous projects [34, 38]. Furthermore, ProtT5 does not require splitting long sequences, which might remove valuable global context information, while ESM-1b can only handle sequences of up to 1022 residues.

Model architecture

Our TMbed model architecture contained three modules (Additional file 1: Fig. S1): a convolutional neural network (CNN) to generate per-residue predictions, a Gaussian smoothing filter, and a Viterbi decoder to find the best class label for each residue. We implemented the model in PyTorch [59].

Module 1: CNN

The first component of TMbed is a CNN with four layers (Additional file 1: Fig. S1). The first layer is a pointwise convolution, i.e., a convolution with kernel size of 1, which reduces the ProtT5 embeddings for each residue (position in the sequence) from 1024 to 64 dimensions. Next, the model applies layer normalization [60] along the sequence and feature dimensions, followed by a ReLU (Rectified Linear Unit) activation function to introduce non-linearity. The second and third layers consist of two parallel depthwise convolutions; both process the output of the first layer. As depthwise convolutions process each input dimension (feature) independently while considering consecutive residues, those two layers effectively generate sliding weighted sums for each dimension. The kernel sizes of the second and third layer are 9 and 21, respectively, corresponding

to the average length of transmembrane beta strands and helices. As before, the model normalizes the output of both layers and applies the ReLU function. It then concatenates the output of all three layers, constructing a 192-dimensional feature vector for each residue (position in the sequence). The fourth layer is a pointwise convolution combining the outputs from the previous three layers and generates scores for each of the five classes: transmembrane beta strand (B), transmembrane helix (H), signal peptide (S), non-membrane inside (i), and non-membrane outside (o).

Module 2: Gaussian filter

This module smooths the output from the CNN for adjacent residues (sequence positions) to reduce noisy predictions. The filter allows flattening isolated single-residue peaks. For instance, peaks extending of only one to three residues for the classes B and H are often non-informative; similarly short peaks for class S are unlikely correct. The filter uses a Gaussian distribution with standard deviation of 1 and a kernel size of 7, i.e., its seven weights correspond to three standard deviation intervals to the left and right, as well as the central peak. A softmax function then converts the filtered class scores to a class probability distribution.

Module 3: Viterbi decoder

The Viterbi algorithm decodes the class probabilities and assigns a class label to each residue (position in the sequence; Additional file 1: Note S3, Fig. S2). The algorithm uses no trainable parameter; it scores transitions according to the predicted class probabilities. Its purpose is to enforce a simple grammar such that (1) signal peptides can only start at the N-terminus (first residue in protein), (2) signal peptides and transmembrane segments must be at least five residues long (a reasonable trade-off between filtering out false positives and still capturing weak signals), and (3) the prediction for the inside/outside orientation has to change after each transmembrane segment (to simulate crossing through the membrane). Unlike the Gaussian filter, we did not apply the Viterbi decoder during training. This simplified backpropagation and sped up training.

Training details

We performed a stratified five-fold nested cross-validation for model development (Additional file 1: Fig. S3). First, we separated our protein sequences into four groups: beta barrel TMPs, alpha helical TMPs with only a single helix, those with multiple helices, and non-membrane proteins. We further subdivided each group into proteins with and without signal peptides. Next, we randomly and evenly distributed all eight groups into five data sets. As all of our data sets were redundancy reduced, no two splits contained similar protein sequences for any of the classes. However, similarities between proteins of two different classes were allowed, not the least to provide more conservative performance estimates.

During development, we used four of the five splits to create the model and the fifth for testing (Additional file 1: Fig. S3). Of the first four splits, we used three to train the model and the fourth for validation (optimize hyperparameters). We repeated this 3–1 split three more times, each time using a different split for the validation set, and calculated the average performance for every hyperparameter configuration. Next, we trained

a model with the best configuration on all four development splits and estimated its final performance on the independent test split. We performed this whole process a total of five times, each time using a different of the five splits as test data and the remaining four for the development data. This resulted in five final models; each trained, optimized, and tested on independent data sets.

We applied weight decay to all trained weights of the model and added a dropout layer right before the fourth convolutional layer, i.e., the output layer of the CNN. For every training sample (protein sequence), the dropout layer randomly sets 50% of the features to zero across the entire sequence, preventing the model from relying on only a specific subset of features for the prediction.

We trained all models for 15 epochs using the AdamW [61] optimizer and cross-entropy loss. We set the beta parameters to 0.9 and 0.999, used a batch size of 16 sequences, and applied exponential learning rate decay by multiplying the learning rate with a factor of 0.8 every epoch. The initial learning rate and weight decay values were part of the hyperparameters optimized during cross-validation (Additional file 1: Table S2).

The final TMbed model constitutes an ensemble over the five models obtained from the five outer cross-validation iterations (Additional file 1: Fig. S3), i.e., one for each training/test set combination. During runtime, each model generates its own class probabilities (CNN, plus Gaussian filter), which are then averaged and processed by the Viterbi decoder to generate the class labels.

Evaluation and other methods

We evaluated the test performance of TMbed on a per-protein level and on a per-segment level (Additional file 1: Note S1). For protein level statistics, we calculated recall and false positive rate (FPR). We computed those statistics for three protein classes: alpha helical TMPs, beta barrel TMPs, and globular proteins.

We distinguished correct and incorrect segment predictions using two constraints: (1) the observed and predicted segment must overlap such that the intersection of the two is at least half of their union, and (2) neither the start nor the end positions may deviate by more than five residues between the observed and predicted segment (Additional file 1: Fig. S4). All segments predicted meeting both these criteria were considered as "correctly predicted segments", all others as "incorrectly predicted segments". This allowed for a reasonable margin of error regarding the position of a predicted segment, while punishing any gaps introduced into a segment. For per-segment statistics, we calculated recall and precision. We also computed the percentage of proteins with the correct number of predicted segments (Q_{num}), the percentage of proteins for which all segments are correctly predicted (Q_{ok}), and the percentage of correctly predicted segments that also have the correct orientation within the membrane (Q_{top}). We considered only proteins that actually contain the corresponding type of segment when calculating per-segment statistics, e.g., only beta barrel TMPs for transmembrane beta strand segments.

We compared TMbed to other prediction methods for alpha helical and beta barrel TMPs (details in Additional file 1: Note S2): BetAware-Deep [15], BOCTOPUS2 [16], CCTOP [17, 18], DeepTMHMM [44], HMM-TM [19–21], OCTOPUS [22], Philius [23], PolyPhobius [24], PRED-TMBB2 [20, 21, 25], PROFtmb [3], SCAMPI2 [26],

SPOCTOPUS [27], TMSEG [28], and TOPCONS2 [29]. We chose those methods based on their good prediction accuracy and public popularity. For methods predicting only either alpha helical or beta barrel TMPs, we considered the corresponding other type of TMPs as globular proteins for the per-protein statistics. In addition, we generated signal peptide predictions with SignalP 6.0 [52]. The performance of older TMH prediction methods could be triangulated based on previous comprehensive estimate of such methods [28, 62].

Unless stated otherwise, all reported performance values constitute the average performance over the five independent test sets during cross-validation (c.f. *Training details*) and their error margins reflect the 95% confidence interval (CI), i.e., 1.96 times the sample standard error over those five splits (Additional file 1: Tables S5, S6). We considered two values A and B statistically significantly different if they differ by more than their composite 95% confidence interval:

$$|A - B| > CI_c = \sqrt{CI_A^2 + CI_B^2} \tag{1}$$

Additional out-of-distribution benchmark

In the most general sense, machine learning models learn and predict distributions. Most membrane data sets are small and created using the same resources, including OPM [45], PDBTM [49–51], and UniProtKB/Swiss-Prot [46] that often mix experimental annotations with sophisticated algorithms [50, 63–65] to determine the boundaries of transmembrane segments, e.g., by using the 3D structure. Given these constraints, we might expect data sets from different groups to render similar results. Analyzing the validity of this assumption, we included the data set assembled for the development of DeepTMHMM [44]. Three reasons made us chose this set as an alternative perspective: (1) it is recent, (2) it contains helical and beta barrel TMPs, and (3) the authors made their cross-validation predictions available, simplifying comparisons.

We created two distinct data sets from the DeepTMHMM data. First, we collected all proteins common to both data sets (TMbed and DeepTMHMM). We used those proteins to estimate how much the annotations within both data sets agree with each other. In total, there were 1788 proteins common to both data sets: 43 beta barrel TMPs, 184 alpha helical TMPs, 1,560 globular proteins, and one protein (MSPA_MYCS2; Porin MspA) which sits in the outer-membrane of *Mycobacterium smegmatis* [66]. We classified this as beta barrel TMP while DeepTMHMM listed it, most likely incorrectly, as a globular protein. The second data set that we created contained all proteins from the DeepTMHMM data set that were non-redundant to the training data of TMbed. We used PSI-BLAST [67] to find all significant (e-value $< 10^{-4}$) local alignments with a 20% PIDE threshold and 40% alignment coverage to remove the redundant sequences. This second data set contained 667 proteins: 14 beta barrel TMPs, 86 alpha helical TMPs, and 567 globular proteins. We generated predictions with TMbed for those proteins and compared them to the cross-validation predictions for DeepTMHMM, as well as the best performing methods from our own benchmark (CCTOP [17, 18], TOPCONS2 [29], BOCTOPUS2 [16]); we used the DeepTMHMM data set annotations as ground truth.

Data set of new membrane proteins

In order to perform a CASP-like performance evaluation, we gathered all PDB structures published since Feb 05, 2022, which is just after the data for our set and that of DeepTMHMM [44] have been collected. This comprised 1,511 PDB structures (more than 250 of which related to the SARS-CoV-2 protein P0DTD1) that we could map to 1,078 different UniProtKB sequences. We then used PSI-BLAST to remove all sequences similar to our data set or that of DeepTMHMM (e-value $< 10^{-4}$, 20% PIDE at 40% coverage), which resulted in 333 proteins. Next, we predicted transmembrane segments within those proteins using TMbed and DeepTMHMM. For 38 proteins, either TMbed or DeepTMHMM predicted transmembrane segments. After removing any sequences shorter than 100 residues (i.e., fragments) and those in which the predicted segments were not within the resolved regions of the PDB structure, we were left with a set of 5 proteins: one beta barrel TMP and four alpha helical TMPs. Finally, we used the PPM [63–65] algorithm from OPM [45] to estimate the actual membrane boundaries.

Results and discussion

We have developed a new machine learning model, dubbed TMbed; it exclusively uses embeddings from the ProtT5 [34] pLM as input to predict for each residue in a protein sequence to which of the following four "classes" it belongs: transmembrane beta strand (TMB), transmembrane helix (TMH), signal peptide (SP), or non-transmembrane segment. It also predicts the inside/outside orientation of TMBs and TMHs within the membrane, indicating which parts of a protein are inside or outside a cell or compartment. Although the prediction of signal peptides was primarily integrated to improve TMH predictions by preventing the confusion of TMHs with SPs and vice versa, we also evaluated and compared the performance for SP prediction of TMbed to that of other methods.

Reaching SOTA in protein sorting

TMbed detected TMPs with TMHs and TMBs at levels similar or numerically above the best state-of-the-art (SOTA) methods that use evolutionary information from multiple sequence alignments (MSA; Table 1: Recall). Compared to MSA-based methods, TMbed achieved this parity or improvement at a significantly lower false positive rate (FPR), tied only with DeepTMHMM [44], another embedding-based method (Table 1: FPR). Given those numbers, we expect TMbed to misclassify only about 215 proteins for a proteome with 20,000 proteins (Additional file 1: Table S10), e.g., the human proteome, while the other methods would make hundreds more mistakes (DeepTMHMM: 331, TOPCONS2: 683, BOCTOPUS2: 880). Such low FPRs suggest our method as an automated high-throughput filter for TMP detection, e.g., for the creation and annotation of databases, or the decision which AlphaFold2 [11, 68] predictions to parse through advanced software annotating transmembrane regions in 3D structures or predictions [45, 49, 69]. In the binary prediction of whether or not a protein has a signal peptide, TMbed achieved similar levels as the specialist SignalP 6.0 [52] and as DeepTMHMM [44], reaching 99% recall at 0.1% FPR (Additional file 1: Table S3).

Table 1 Per-protein performance. *

	β-TMP (57)		α-TMP (571)		Globular (5654)	
	Recall (%)	**FPR (%)**	**Recall (%)**	**FPR (%)**	**Recall (%)**	**FPR (%)**
TMbed	**93.8 ± 7.5**	***0.1 ± 0.1***	**97.5 ± 0.7**	***0.5 ± 0.2***	***99.5 ± 0.2***	2.8 ± 1.2
DeepTMHMM	77.9 ± 12.7	***0.1 ± 0.1***	95.8 ± 1.3	***0.5 ± 0.2***	***99.5 ± 0.2***	5.9 ± 2.2
TMSEG	–	–	96.5 ± 1.0	2.3 ± 0.3	97.7 ± 0.3	3.5 ± 1.0
TOPCONS2[1]	–	–	94.2 ± 1.3	2.6 ± 0.3	97.4 ± 0.3	5.8 ± 1.3
OCTOPUS[1]	–	–	94.2 ± 1.9	9.1 ± 0.7	90.9 ± 0.7	5.8 ± 1.9
Philius[1]	–	–	92.5 ± 1.4	2.6 ± 0.2	97.4 ± 0.2	7.5 ± 1.4
PolyPhobius[1]	–	–	97.2 ± 1.1	5.3 ± 0.4	94.7 ± 0.4	2.8 ± 1.1
SPOCTOPUS[1]	–	–	**97.5 ± 1.6**	17.2 ± 0.8	82.8 ± 0.8	**2.5 ± 1.6**
SCAMPI2 (MSA)	–	–	94.2 ± 1.6	5.6 ± 0.3	94.4 ± 0.3	5.8 ± 1.6
CCTOP[2]			96.1 ± 2.1	3.7 ± 0.6	96.3 ± 0.6	3.9 ± 2.1
HMM-TM (MSA)[3]	–	–	97.3 ± 1.6	21.4 ± 0.5	78.6 ± 0.5	2.7 ± 1.6
BOCTOPUS2	84.0 ± 13.3	4.2 ± 0.5	–	–	95.8 ± 0.5	16.0 ± 13.3
BetAware-Deep	85.1 ± 9.3	4.7 ± 0.3	–	–	95.3 ± 0.3	14.9 ± 9.3
PRED-TMBB2[4]	88.8 ± 12.1	7.1 ± 0.4	–	–	92.9 ± 0.4	11.2 ± 12.1
PROFtmb	91.9 ± 9.0	6.1 ± 0.5	–	–	93.9 ± 0.5	8.1 ± 9.0

*Evaluation of the ability to distinguish between 57 beta barrel TMPs (β-TMP), 571 alpha helical TMPs (α-TMP) and 5654 globular, water-soluble non-TMP proteins in our data set. Recall and false positive rate (FPR) were averaged over the five independent cross-validation test sets; error margins given for the 95% confidence interval (1.96*standard error); bold: best values for each column; italics: differences statistically significant with over 95% confidence (only computed between best and 2nd best, or all methods ranked 1 and those ranked lower)

[1] Evaluation missing for one of 5,654 globular proteins

[2] Evaluation missing for one of 571 α-TMPs and six of 5,654 globular proteins

[3] Evaluation includes only 51 β-TMPs, 552 α-TMPs, and 5,524 globular proteins due to runtime errors

[4] The local PRED-TMBB2 version did not include the pre-filtering step of the web server. This caused a FPR for β-TMP of almost 78%. Thus, we listed the statistics for the web server predictions, which did not include MSA input

Many of the beta barrel TMPs that prediction methods missed had only two or four transmembrane beta strands (TMB). Such proteins cannot form a pore on their own, instead they have to form complexes with other proteins to function as TMPs, either by binding to other proteins or by forming multimers with additional copies of the same proteins by, e.g., trimerization. In fact, all four beta barrel TMPs missed by TMbed fell into this category. Thus, as all other methods, TMbed performed, on average, worse for beta barrel TMPs that cannot form pores alone. This appeared unsurprising, as the input to all methods were single proteins. For TMPs with TMHs, we also observed lower performance in the distinction between TMP/other for TMPs with a single TMH (recall: 93 ± 3%) compared to those with multiple TMHs (recall: 99 ± 1%). However, TMPs with single helices can function alone.

The embedding-based methods TMbed (introduced here using ProtT5 [34]) and DeepTMHMM [44] (based on ESM-1b [36]) performed at least on par with the SOTA using evolutionary information from MSA (Table 1). While this was already impressive, the real advantage was in the speed. For instance, our method, TMbed, predicted all 6,517 proteins in our data set in about 13 min (i.e., about eight sequences per second) on our server machine (Additional file 1: Table S1); this runtime included generating the ProtT5 embeddings. The other embedding-based method, DeepTMHMM, needed about twice as long (23 min). Meanwhile, methods that search databases and

Table 2 Per-segment performance for TMH (transmembrane helices). *

	TMH (571/2936)				
	Recall (%)	**Precision (%)**	**Q_{ok} (%)**	**Q_{num} (%)**	**Q_{top} (%)**
TMbed	***88.7 ± 0.6***	***88.7 ± 0.7***	***62.4 ± 3.7***	**86.0 ± 2.3**	**96.4 ± 2.7**
DeepTMHMM	80.0 ± 2.4	80.5 ± 2.4	46.2 ± 4.8	85.7 ± 3.5	96.3 ± 2.2
TMSEG	74.5 ± 2.4	77.1 ± 1.7	35.6 ± 2.4	69.9 ± 2.7	83.8 ± 4.7
TOPCONS2	76.4 ± 1.5	78.4 ± 0.8	41.0 ± 3.1	74.4 ± 3.3	91.7 ± 3.1
OCTOPUS	71.6 ± 1.5	75.7 ± 1.4	36.0 ± 2.8	67.6 ± 3.4	87.5 ± 3.1
Philius	70.8 ± 2.2	73.7 ± 0.8	34.2 ± 3.7	66.9 ± 3.4	87.5 ± 2.9
PolyPhobius	76.0 ± 2.1	76.4 ± 1.1	40.3 ± 3.5	74.5 ± 2.8	86.8 ± 2.7
SPOCTOPUS	71.5 ± 1.2	75.8 ± 1.2	35.7 ± 3.3	67.4 ± 5.5	87.2 ± 3.4
SCAMPI2 (MSA)	72.3 ± 2.7	74.1 ± 1.5	33.5 ± 3.0	72.2 ± 4.5	90.6 ± 3.5
CCTOP[1]	77.0 ± 1.7	79.4 ± 1.0	41.9 ± 3.6	82.6 ± 2.7	92.6 ± 2.6
HMM-TM (MSA)[2]	73.3 ± 1.7	72.5 ± 1.2	33.5 ± 1.4	72.1 ± 3.0	88.3 ± 4.2

*Segment performance for transmembrane helix (TMH) prediction based on 571 alpha helical TMPs (α-TMP) with a total of 2936 TMHs. Recall, Precision, Q_{ok}, Q_{num}, and Q_{top} were averaged over the five independent cross-validation test sets; error margins given for the 95% confidence interval (1.96*standard error); bold: best values for each column; italics: differences statistically significant with over 95% confidence (only computed between best and 2nd best).

[1] Evaluation missing for one of 571 α-TMPs.

[2] Evaluation includes only 552 of the 571 α-TMPs due to runtime errors of the method.

Table 3 Per-segment performance for TMB (transmembrane beta strands). *

	TMB (57/768)				
	Recall (%)	**Precision (%)**	**Q_{ok} (%)**	**Q_{num} (%)**	**Q_{top} (%)**
TMbed	**95.0 ± 4.3**	***99.2 ± 0.7***	***80.5 ± 11.4***	**88.1 ± 6.9**	**98.1 ± 3.8**
DeepTMHMM	85.9 ± 6.6	92.5 ± 4.7	46.1 ± 7.6	74.3 ± 13.0	97.2 ± 4.4
BOCTOPUS2	85.3 ± 9.2	96.6 ± 2.0	56.6 ± 18.9	71.2 ± 11.8	98.0 ± 2.0
BetAware-Deep	67.1 ± 6.5	62.2 ± 11.4	8.7 ± 5.3	60.9 ± 14.1	95.7 ± 5.4
PRED-TMBB2 (MSA)	85.4 ± 1.9	75.6 ± 4.8	18.4 ± 15.0	44.5 ± 26.7	95.9 ± 3.4
PROFtmb	78.2 ± 10.1	78.0 ± 6.9	20.2 ± 12.8	46.6 ± 11.7	97.2 ± 1.0

*Segment performance for transmembrane beta strand (TMB) prediction based on 57 beta barrel TMPs (β-TMP) with a total of 768 TMBs. Recall, Precision, Q_{ok}, Q_{num}, and Q_{top} were averaged over the five independent cross-validation test sets; error margins given for the 95% confidence interval (1.96*standard error); bold: best values for each column; italics: differences statistically significant with over 95% confidence (only computed between best and 2nd best)

generate MSAs usually take several seconds or minutes for a single protein sequence [70], or require significant amounts of computing resources (e.g., often more than 100 GB of memory) to achieve comparable runtimes [55].

Excellent transmembrane segment prediction performance

TMbed reached the highest performance for transmembrane segments amongst all methods evaluated (Tables 2, 3). With recall and precision values of 89 ± 1% for TMHs, it significantly outperformed the second best and only other embedding-based method, DeepTMHMM, (80 ± 2%, Table 2). TMbed essentially predicted 62% of all transmembrane helical (TMH) TMPs completely correctly (Q_{ok}, i.e., all TMHs within ± 5 residues of true annotation). DeepTMHMM reached second place with Q_{ok} of 46 ± 4%. This difference between TMbed and DeepTMHMM was over twice that between

DeepTMHMM and the two methods performing third best by this measure, CCTOP [17, 18] and TOPCONS2 [29], which are based on evolutionary information.

The results were largely similar for beta barrel TMPs (TMBs) with TMbed achieving the top performance by all measures: reaching 95% recall and an almost perfect 99% precision. The most pronounced difference was a 23 percentage points lead in Q_{ok} with 80%, compared to BOCTOPUS2 [16] with 57% in second place. Overall, TMbed predicted the correct number of transmembrane segments in 86–88% of TMPs and correctly oriented 98% of TMBs and 96% of TMHs. For signal peptides, TMbed performed on par with SignalP 6.0, reaching 93% recall and 95% precision (Additional file 1: Table S3). For this task, both methods appeared to be slightly outperformed by DeepTMHMM. However, none of those differences exceeded the 95% confidence interval, i.e., the numerically consistent differences were not statistically significant. On top, the signal peptide expert method SignalP 6.0 is the only of the three that distinguishes between different types of signal peptides.

As for the overall per-protein distinction between TMP and non-TMP, the per-segment recall and precision also slightly correlated with the number of transmembrane segments, i.e., the more TMHs or TMBs in a protein the higher the performance (Additional file 1: Table S4). Again, as for the TMP/non-TMP distinction, beta barrel TMPs with only two or four TMBs differed most to those with eight or more.

Gaussian filter and Viterbi decoder improve segment performance

TMbed introduced a Gaussian filter smoothing over some local peaks in the prediction and a Viterbi decoder implicitly enforcing some "grammar-like" rules (Materials & Methods). We investigated the effect of these concepts by comparing the final TMbed architecture to three simpler alternatives: one variant used only the CNN, the other two variants combined the simple CNN with either the Gaussian filter or the Viterbi decoder, not both as TMbed. For the variants without the Gaussian filter, we retrained the CNN using the same hyperparameters but without the filter. Individually, both modules (filter and decoder) significantly improved precision and Q_{ok} for both TMH and TMB, while recall remained largely unaffected (Additional file 1: Table S9). Clearly, either step already improved over just the CNN. However, which of the two was most important depended on the type of TMP: for TMH proteins Viterbi decoder mattered more, for TMB proteins the Gaussian filter. Both steps together performed best throughout without adding any significant overhead to the overall computational costs compared to the other components.

Self-predictions reveal potential membrane proteins

We checked for potential overfitting of our model by predicting the complete data set with the final TMbed ensemble. This meant that four of the five models had seen each of those proteins during training. While the number of misclassified proteins went down, we found that there were still some false predictions, indicating that our models did not simply learn the training data by heart (Additional file 1: Tables S7, S8). In fact, upon closer inspection of the 11 false positive predictions (8 alpha helical and 3 beta barrel TMPs), those appear to be transmembrane proteins incorrectly classified as globular proteins in our data set due to missing annotations in UniProtKB/Swiss-Prot, rather

than incorrect predictions. Two of them, P09489 and P40601, have automatic annotations for an autotransporter domain, which facilitates transport through the membrane. Further, we processed the predicted AlphaFold2 [11, 68] structures of all 11 proteins using the PPM [45] algorithm, which tries to embed 3D structures into a membrane bilayer. For eight of those, the predicted transmembrane segments correlated well with the predicted 3D structures and membrane boundaries (Fig. 1; Additional file 1: Fig. S5). For the other three, the 3D structures and membrane boundaries still indicate transmembrane domains within those proteins, but the predicted transmembrane segments only cover parts of those domains (Additional file 1: Fig. S5, last row). Together, these predictions provided convincing evidence for considering all eleven proteins as TMPs.

Predicting the human proteome in less than an hour

Given that our new method already outperformed the SOTA using evolutionary information from MSAs, the even more important advantage was speed. To estimate prediction throughput, we applied TMbed to all human proteins in 20,375 UniProtKB/Swiss-Prot (version: April 2022; excluding TITIN_HUMAN due to its extreme length of 34,350 residues). Overall, it took our server machine (Additional file 1: Table S1) only 46 min to generate all embeddings and predictions (estimate for consumer-grade PC in the next section). TMbed identified 14 beta barrel TMPs and 4,953 alpha helical TMPs, matching previous estimates for alpha helical TMPs [1, 28]. Two of the 14 TMBs appear to be false positives as TMbed predicted only a single TMB in each protein. The other 12 proteins are either part of the Gasdermin family (A to E), or associated with the mitochondrion, including three proteins for a voltage-dependent anion-selective channel and the TOM40 import receptor.

Further, we generated predictions for all proteins from UniProtKB/Swiss-Prot (version: May 2022), excluding sequences above 10,000 residues (20 proteins). Processing those 566,976 proteins took about 8.5 h on our server machine. TMbed predicted 1,702 beta barrel TMPs and 77,296 alpha helical TMPs (predictions available via our GitHub repository).

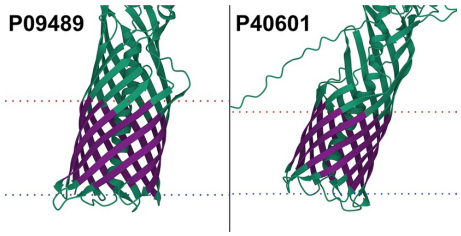


Fig. 1 Potential transmembrane proteins in the globular data set. AlphaFold2 [11, 68] structure of extracellular serine protease (P09489) and Lipase 1 (P40601). Transmembrane segments (dark purple) predicted by TMbed correlate well with membrane boundaries (dotted lines: red = outside, blue = inside) predicted by the PPM [45] web server. Images created using Mol* Viewer [71]. Though our data set lists them as globular proteins, the predicted structures indicate transmembrane domains, which align with segments predicted by our method. The predicted domains overlap with autotransporter domains detected by the UniProtKB [46] automatic annotation system. Transmembrane segment predictions were made with the final TMbed ensemble model

Hardware requirements

Our model needs about 2.5 GB of memory on the GPU when in 16-bit format. The additional memory needed during inference grows with the square of sequence length due to the attention mechanism of the transformer architecture. On our consumer-grade desktop PC (Additional file 1: Table S1), this translated to a maximum sequence length of about 4,200 residues without maxing out the 12 GB of GPU memory. This barred 76 (0.4%) of the 20,376 human proteins from analysis on a personal consumer-hardware solution (NVIDIA GeForce RTX 3060). The prediction (including embedding generation) for 99.6% of the human proteome (20,376 proteins) took about 57 min on our desktop PC. While it is possible to run the model on a CPU, instead of on a GPU, we do not recommend this due to over tenfold larger runtimes. More importantly, the current lack of support of 16-bit floating-point format on CPUs would imply doubling the memory footprint of the model and computations.

Out-of-distribution performance

The two pLM-based methods DeepTMHMM [44] and TMbed appeared to reach similar performance according to the additional out-of-distribution data set (Additional file 1: Tables S11, S12). While DeepTMHMM reached higher scores for beta barrel proteins (Q_{ok} of $79 \pm 22\%$ vs. $64 \pm 26\%$), these were not quite statistically significant. On the other hand, TMbed managed to outperform DeepTMHMM for alpha helical TMPs (Q_{ok} of $53 \pm 11\%$ vs. $47 \pm 10\%$), though again without statistical significance. Furthermore, TMbed performed on par with the OPM baseline (Additional file 1: Table S12), i.e., using the OPM annotations as predictions for the DeepTMHMM data set, implying that TMbed reached its theoretical performance limit on that data set. Surprisingly, TOPCONS2 and CCTOP both outperformed TMbed and DeepTMHMM with Q_{ok} of $65 \pm 10\%$ and $64 \pm 10\%$ (both not statistically significant), respectively.

Taking a closer look at the length distribution for the transmembrane segments in the TMbed and DeepTMHMM data set annotations and predictions (Additional file 1: Fig. S6) revealed differences. First, while the TMB segments in both data sets averaged 9 residues in length, the DeepTMHMM distribution was slightly shifted toward shorter segments (left in Additional file 1: Fig. S6A) but with a wider spread towards longer segments (right in Additional file 1: Fig. S6A). Both of these features were mirrored in the distribution of predicted TMBs. In contrast, the TMH distributions for DeepTMHMM showed an unexpected peak for TMH with 21 residues (both in the annotations used to train DeepTMHMM and in the predictions). In fact, the peak for annotated TMHs at 21 was more than double the value of the two closest length-bins (TMH = 20|22) combined. As the lipid bilayer remains largely invisible in X-ray structures, the exact begin and ends of TMHs may have some errors [28, 45, 49–51, 62]. Thus, when plotting the distribution of TMH length, we expected some kind of normal distribution with a peak around 20-odd residues with more points for longer than for shorter TMHs [72]. In stark contrast to this expectation, the distribution observed for the TMHs used to develop DeepTMHMM appeared to have been obtained through some very different protocol (Additional file 1: Fig. S6B).

In contrast, the distributions for the annotations from OMP and the predictions from TMbed appeared to be more normally distributed with TMH lengths exhibiting a slight

peak at 22 residues. The larger the AI model, the more it succeeds in reproducing features of the development set even when those might be based on less experimentally supported aspects. The DeepTMHMM model reproduced the dubious experimental distribution of TMHs exceedingly (Additional file 1: Fig. S6B, e.g., orange line and bars around peak at 16). Although we do not know the origin of this bias in the DeepTMHMM data set, we have seen similar bias in some prediction methods and automated annotations in UniProtKB/Swiss-Prot. In fact, a quick investigation showed that for 80 of the 184 common alpha helical TMPs the DeepTMHMM annotations matched those found in UniProtKB but not the OPM annotation in our TMbed data set. Of those annotations, 66% (303 of 459) were 21-residues long TMHs, accounting for 73% of all such segments; the other 104 TMPs contained only 19% (114 of 593) TMHs of length 21. This led us to believe that the DeepTMHMM data set contained, in part, length-biased annotations found in UniProtKB. Other examples of methods with length biases include SCAMPI2 and TOPCONS2 that both predicted exclusively TMHs with 21 residues; OCTOPUS and SPOCTOPUS predicted only TMHs of length 15, 21, and 31 (with more than 90% of those being 21 residues). BOCTOPUS2 predicted only beta strands of length 8, 9, and 10, with about 80% of them being nine residues long.

Since TMHs are around 21 residues long, such bias is not necessarily relevant. However, it might point to why performance appears better against some data sets supported less by high-resolution experiments than by others.

Performance on new membrane proteins

Although, the small data set size did not allow for statistically significant results (Additional file 1: Table S13), TMbed performed numerically better than the other methods; in particular, BOCTOPUS2 failed to predict the only beta barrel TMP. While TMbed and DeepTMHMM both missed two of the 30 transmembrane beta strands, TMbed placed the remaining ones, on average, more accurately (recall: 93% vs 87%; precision: 100% vs. 93%). All methods performed worse for the alpha helical TMPs than on the other two benchmark data set, though with a sample size of only four proteins (25 TMHs total), we cannot be sure if this is an effect of testing on novel membrane proteins or simply by chance. Nevertheless, the transmembrane segments predicted by TMbed fit quite well to the membrane boundaries estimated by the PPM [63–65] algorithm (Fig. 2).

No data leakage through pLM

pLMs such as ProtT5 [34] used by TMbed or ESM-1b [36] used by DeepTMHMM are pre-trained on billions of protein sequences. Typically, these include all protein sequences known today. In particular, they include all membrane and non-membrane proteins used in this study. In fact, assuming that the TMPs of known structure account for about 2–5% [78, 79] of all TMPs and that TMPs account for about 20–25% of all proteins, we assume pLMs have been trained on over 490 million TMPs that remain to be experimentally characterized. For the development of AI/ML solutions, it is crucial to establish that methods do not over-fit to existing data but that they will also work for new, unseen data. This implies that in the standard cross-validation process, it is important to not leak any data from development (training and validation used for hyperparameter optimization and model choice)

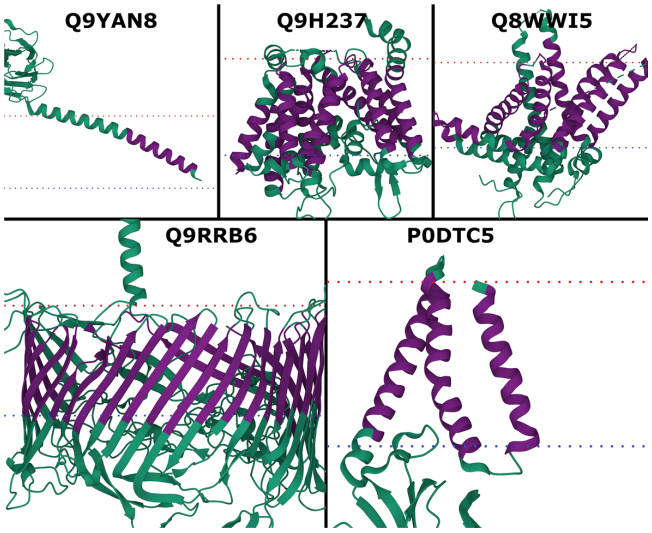


Fig. 2 New membrane proteins. PDB structures for probable flagellin 1 (Q9YAN8; 7TXI [73]), protein-serine O-palmitoleoyltransferase porcupine (Q9H237; 7URD [74]), choline transporter-like protein 1 (Q8WWI5; 7WWB [75]), S-layer protein SlpA (Q9RRB6; 7ZGY [76]), and membrane protein (P0DTC5; 8CTK [77]). Transmembrane segments (dark purple) predicted by TMbed; membrane boundaries (dotted lines: red = outside, blue = inside) predicted by the PPM [45] web server. Images created using Mol* Viewer [71]

to test set (used to assess performance). This implies the necessity for redundancy reduction. This also implies that the conditions for the test set are exactly the same as those that will be encountered in future predictions. For instance, if today's experimental annotations were biased toward bacterial proteins, we might expect performance to be worse for eukaryotic proteins and vice versa.

Both TMbed introduced here and DeepTMHMM are based on the embeddings of pre-trained pLMs; both accomplish the TM-prediction through a subsequent step dubbed transfer learning, in which they use the pLM embeddings as input to train a new AI/ML model in supervised manner on some annotations about membrane segments. Could any data leak from the training of pLMs into the subsequent step of training the TM-prediction methods? Strictly speaking, if no experimental annotations are used, no annotations can leak: the pLMs used here never saw any annotation other than protein sequences.

Even when no annotations could have leaked because none were used for the pLM, should we still ascertain that the conditions for the test set and for the protein for which the method will be applied in the future are identical? We claim that we do not have to ascertain this. However, we cannot support any data for (nor against) this claim. To play devil's advocate, let us assume we had to. The reality is that the vast majority of all predictions likely to be made over the next five years will be for proteins included in these pLMs. In other words, the conditions for future use-cases are exactly the same as those used in our assessment.

Conclusions

TMbed predicts alpha helical (TMH) and beta barrel (TMB) transmembrane proteins (TMPs) with high accuracy (Table 1), performing at least on par or even better than state-of-the-art (SOTA) methods, which depend on evolutionary information from multiple sequence alignments (MSA; Tables 1, 2, 3). In contrast, TMbed exclusively inputs sequence embeddings from the protein language model (pLM) ProtT5. Our novel method shines, in particular, through its low false positive rate (FPR; Table 1), incorrectly predicting fewer than 1% of globular proteins to be TMPs. TMbed also numerically outperformed all other tested methods in terms of correctly predicting transmembrane segments (on average, 9 out of 10 segments were correct; Tables 2, 3). Despite its top performance, the even more significant advantage of TMbed is speed: the high throughput rate of the ProtT5 [34] encoder enables predictions for entire proteomes within an hour, given a suitable GPU (Additional file 1: Table S1). On top, the method runs on consumer-grade GPUs as found in more recent gaming and desktop PCs. Thus, TMbed can be used as a proteome-scale filtering step to scan for transmembrane proteins. Validating the predicted segments with AlphaFold2 [11, 68] structures and the PPM [45] method could be combined into a fast pipeline to discover new membrane proteins, as we have demonstrated with a few proteins. Finally, we provide predictions for 566,976 proteins from UniProtKB/Swiss-Prot (version: May 2022) via our GitHub repository.

Abbreviations

CI	Confidence interval
CNN	Convolutional neural network
MSA	Multiple sequence alignment
OPM	Orientations of proteins in membranes database
PDB	Protein data bank
PDBTM	Protein data bank of transmembrane proteins
pLM	Protein language model
SIFTS	Structure integration with function, taxonomy and sequence
SOTA	State-of-the-art
SP	Signal peptide
TMB	Transmembrane beta strand
TMH	Transmembrane helix
TMP	Transmembrane protein

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-04873-x.

Additional file 1. Supporting Online Material (SOM) containing additional figures, tables and notes.

Acknowledgements

Thanks to Tim Karl and Inga Weise for their help with technical and administrative issues; to Tobias Olenyi, Michael Heinzinger, and Christian Dallago for thoughtful discussions, help with ProtT5, and help with the manuscript; to Konstantinos Tsirigos and Ioannis Tamposis for their support with setting up HMM-TM and PRED-TMBB2; to Pier Luigi Martelli for providing us with BetAware-Deep predictions. Thanks to all who deposit their experimental data in public databases, and to those who maintain them. Last but not least, we thank the reviewers for their constructive criticism, which helped to improve our manuscript.

Author contributions

MB collected the data sets, developed and evaluated the TMbed model, and took the lead in writing the manuscript. BR supervised and guided the work, and co-wrote the manuscript. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. The server machine to run the ProtT5 model was funded by Software Campus Funding (BMBF 01IS17049).

Availability of data and materials

Our code, method, and data sets are freely available in the GitHub repository, https://github.com/BernhoferM/TMbed.

Declarations

Ethical approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 12 June 2022 Accepted: 3 August 2022
Published online: 08 August 2022

References

1. Fagerberg L, Jonasson K, von Heijne G, Uhlen M, Berglund L. Prediction of the human membrane proteome. Proteomics. 2010;10(6):1141–9.
2. Liu J, Rost B. Comparing function and structure between entire proteomes. Protein Sci. 2001;10(10):1970–9.
3. Bigelow HR, Petrey DS, Liu J, Przybylski D, Rost B. Predicting transmembrane beta-barrels in proteomes. Nucleic Acids Res. 2004;32(8):2566–77.
4. Overington JP, Al-Lazikani B, Hopkins AL. How many drug targets are there? Nat Rev Drug Discov. 2006;5(12):993–6.
5. von Heijne G. The membrane protein universe: what's out there and why bother? J Intern Med. 2007;261(6):543–57.
6. ww PDBc. Protein Data Bank: the single global archive for 3D macromolecular structure data. Nucleic Acids Res. 2019;47(D1):D520–D8.
7. Berman H, Henrick K, Nakamura H. Announcing the worldwide Protein Data Bank. Nat Struct Biol. 2003;10(12):980.
8. Hendrickson WA. Atomic-level analysis of membrane-protein structure. Nat Struct Mol Biol. 2016;23(6):464–7.
9. Varga J, Dobson L, Remenyi I, Tusnady GE. TSTMP: target selection for structural genomics of human transmembrane proteins. Nucleic Acids Res. 2017;45(D1):D325–30.
10. Newport TD, Sansom MSP, Stansfeld PJ. The MemProtMD database: a resource for membrane-embedded protein structures and their lipid interactions. Nucleic Acids Res. 2019;47(D1):D390–7.
11. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596(7873):583–9.
12. Marx V. Method of the Year: protein structure prediction. Nat Methods. 2022;19(1):5–10.
13. Bordin N, Sillitoe I, Nallapareddy V, Rauer C, Lam SD, Waman VP, et al. AlphaFold2 reveals commonalities and novelties in protein structure space for 21 model organisms. bioRxiv. 2022:2022.06.02.494367.
14. Hegedus T, Geisler M, Lukacs GL, Farkas B. Ins and outs of AlphaFold2 transmembrane protein structure predictions. Cell Mol Life Sci. 2022;79(1):73.
15. Madeo G, Savojardo C, Martelli PL, Casadio R. BetAware-deep: an accurate web server for discrimination and topology prediction of prokaryotic transmembrane beta-barrel proteins. J Mol Biol. 2021;433(11): 166729.
16. Hayat S, Peters C, Shu N, Tsirigos KD, Elofsson A. Inclusion of dyad-repeat pattern improves topology prediction of transmembrane beta-barrel proteins. Bioinformatics. 2016;32(10):1571–3.
17. Dobson L, Remenyi I, Tusnady GE. The human transmembrane proteome. Biol Direct. 2015;10:31.
18. Dobson L, Remenyi I, Tusnady GE. CCTOP: a Consensus Constrained TOPology prediction web server. Nucleic Acids Res. 2015;43(W1):W408–12.
19. Bagos PG, Liakopoulos TD, Hamodrakas SJ. Algorithms for incorporating prior topological information in HMMs: application to transmembrane proteins. BMC Bioinform. 2006;7:189.
20. Tamposis IA, Sarantopoulou D, Theodoropoulou MC, Stasi EA, Kontou PI, Tsirigos KD, et al. Hidden neural networks for transmembrane protein topology prediction. Comput Struct Biotechnol J. 2021;19:6090–7.
21. Tamposis IA, Theodoropoulou MC, Tsirigos KD, Bagos PG. Extending hidden Markov models to allow conditioning on previous observations. J Bioinform Comput Biol. 2018;16(5):1850019.
22. Viklund H, Elofsson A. OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar. Bioinformatics. 2008;24(15):1662–8.
23. Reynolds SM, Kall L, Riffle ME, Bilmes JA, Noble WS. Transmembrane topology and signal peptide prediction using dynamic bayesian networks. PLoS Comput Biol. 2008;4(11):e1000213.
24. Kall L, Krogh A, Sonnhammer EL. An HMM posterior decoder for sequence feature prediction that includes homology information. Bioinformatics. 2005;21(Suppl 1):i251–7.
25. Tsirigos KD, Elofsson A, Bagos PG. PRED-TMBB2: improved topology prediction and detection of beta-barrel outer membrane proteins. Bioinformatics. 2016;32(17):i665–71.
26. Peters C, Tsirigos KD, Shu N, Elofsson A. Improved topology prediction using the terminal hydrophobic helices rule. Bioinformatics. 2016;32(8):1158–62.
27. Viklund H, Bernsel A, Skwark M, Elofsson A. SPOCTOPUS: a combined predictor of signal peptides and membrane protein topology. Bioinformatics. 2008;24(24):2928–9.
28. Bernhofer M, Kloppmann E, Reeb J, Rost B. TMSEG: Novel prediction of transmembrane helices. Proteins. 2016;84(11):1706–16.
29. Tsirigos KD, Peters C, Shu N, Kall L, Elofsson A. The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. Nucleic Acids Res. 2015;43(W1):W401–7.
30. Asgari E, Mofrad MR. Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS ONE. 2015;10(11):e0141287.

31. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM. Unified rational protein engineering with sequence-based deep representation learning. Nat Methods. 2019;16(12):1315–22.
32. Heinzinger M, Elnaggar A, Wang Y, Dallago C, Nechaev D, Matthes F, et al. Modeling aspects of the language of life through transfer-learning protein sequences. BMC Bioinform. 2019;20(1):723.
33. Bepler T, Berger B. Learning the protein language: evolution, structure, and function. Cell Syst. 2021;12(6):654-69 e3.
34. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, et al. ProtTrans: towards cracking the language of Lifes code through self-supervised deep learning and high performance computing. IEEE Trans Pattern Anal Mach Intell. 2021.
35. Ofer D, Brandes N, Linial M. The language of proteins: NLP, machine learning & protein sequences. Comput Struct Biotechnol J. 2021;19:1750–8.
36. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci USA. 2021;118(15):e2016239118.
37. Wu Z, Johnston KE, Arnold FH, Yang KK. Protein sequence design with deep generative models. Curr Opin Chem Biol. 2021;65:18–27.
38. Marquet C, Heinzinger M, Olenyi T, Dallago C, Erckert K, Bernhofer M, et al. Embeddings from protein language models predict conservation and variant effects. Hum Genet. 2021.
39. Littmann M, Heinzinger M, Dallago C, Weissenow K, Rost B. Protein embeddings and deep learning predict binding residues for various ligand classes. Sci Rep. 2021;11(1):23916.
40. Littmann M, Heinzinger M, Dallago C, Olenyi T, Rost B. Embeddings from deep learning transfer GO annotations beyond homology. Sci Rep. 2021;11(1):1160.
41. Sledzieski S, Singh R, Cowen L, Berger B. D-SCRIPT translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. Cell Syst. 2021;12(10):969-82 e6.
42. Heinzinger M, Littmann M, Sillitoe I, Bordin N, Orengo C, Rost B. Contrastive learning on protein embeddings enlightens midnight zone. bioRxiv. 2022:2021.11.14.468528.
43. Weißenow K, Heinzinger M, Rost B. Protein language model embeddings for fast, accurate, alignment-free protein structure prediction. bioRxiv. 2021:2021.07.31.454572.
44. Hallgren J, Tsirigos KD, Pedersen MD, Almagro Armenteros JJ, Marcatili P, Nielsen H, et al. DeepTMHMM predicts alpha and beta transmembrane proteins using deep neural networks. bioRxiv. 2022:2022.04.08.487609.
45. Lomize MA, Pogozheva ID, Joo H, Mosberg HI, Lomize AL. OPM database and PPM web server: resources for positioning of proteins in membranes. Nucleic Acids Res. 2012;40(Database issue):D370–6.
46. UniProt C. UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Res. 2021;49(D1):D480–9.
47. Dana JM, Gutmanas A, Tyagi N, Qi G, O'Donovan C, Martin M, et al. SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. Nucleic Acids Res. 2019;47(D1):D482–9.
48. Velankar S, Dana JM, Jacobsen J, van Ginkel G, Gane PJ, Luo J, et al. SIFTS: structure integration with function, taxonomy and sequences resource. Nucleic Acids Res. 2013;41(Database issue):D483–9.
49. Kozma D, Simon I, Tusnady GE. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. Nucleic Acids Res. 2013;41(Database issue):D524–9.
50. Tusnady GE, Dosztanyi Z, Simon I. Transmembrane proteins in the Protein Data Bank: identification and classification. Bioinformatics. 2004;20(17):2964–72.
51. Tusnady GE, Dosztanyi Z, Simon I. PDB_TM: selection and membrane localization of transmembrane proteins in the protein data bank. Nucleic Acids Res. 2005;33(Database issue):D275–8.
52. Teufel F, Almagro Armenteros JJ, Johansen AR, Gislason MH, Pihl SI, Tsirigos KD, et al. SignalP 6.0 predicts all five types of signal peptides using protein language models. Nat Biotechnol. 2022.
53. Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics. 2012;28(23):3150–2.
54. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics. 2006;22(13):1658–9.
55. Mirdita M, Steinegger M, Soding J. MMseqs2 desktop and local web server app for fast, interactive sequence searches. Bioinformatics. 2019;35(16):2856–8.
56. Papaloukas C, Granseth E, Viklund H, Elofsson A. Estimating the length of transmembrane helices using Z-coordinate predictions. Protein Sci. 2008;17(2):271–8.
57. Granseth E, Viklund H, Elofsson A. ZPRED: predicting the distance to the membrane center for residues in alpha-helical membrane proteins. Bioinformatics. 2006;22(14):e191–6.
58. Nugent T, Jones DT. Transmembrane protein topology prediction using support vector machines. BMC Bioinform. 2009;10:159.
59. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. 2019.
60. Lei Ba J, Kiros JR, Hinton GE. Layer normalization, 2016 July 01, 2016: arXiv:1607.06450. https://ui.adsabs.harvard.edu/abs/2016arXiv160706450L.
61. Loshchilov I, Hutter F. Decoupled weight decay regularization 2017 November 01, 2017. arXiv:1711.05101. https://ui.adsabs.harvard.edu/abs/2017arXiv171105101L.
62. Reeb J, Kloppmann E, Bernhofer M, Rost B. Evaluation of transmembrane helix predictions in 2014. Proteins. 2015;83(3):473–84.
63. Lomize AL, Pogozheva ID, Mosberg HI. Anisotropic solvent model of the lipid bilayer. 2. Energetics of insertion of small molecules, peptides, and proteins in membranes. J Chem Inf Model. 2011;51(4):930–46.
64. Lomize AL, Pogozheva ID, Lomize MA, Mosberg HI. Positioning of proteins in membranes: a computational approach. Protein Sci. 2006;15(6):1318–33.
65. Lomize AL, Todd SC, Pogozheva ID. Spatial arrangement of proteins in planar and curved membranes by PPM 3.0. Protein Sci. 2022;31(1):209–20.

66. Mahfoud M, Sukumaran S, Hulsmann P, Grieger K, Niederweis M. Topology of the porin MspA in the outer membrane of Mycobacterium smegmatis. J Biol Chem. 2006;281(9):5908–15.
67. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25(17):3389–402.
68. Varadi M, Anyango S, Deshpande M, Nair S, Natassia C, Yordanova G, et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res. 2022;50(D1):D439–44.
69. Tunyasuvunakool K, Adler J, Wu Z, Green T, Zielinski M, Zidek A, et al. Highly accurate protein structure prediction for the human proteome. Nature. 2021;596(7873):590–6.
70. Bernhofer M, Dallago C, Karl T, Satagopam V, Heinzinger M, Littmann M, et al. PredictProtein—predicting protein structure and function for 29 years. Nucleic Acids Res. 2021;49(W1):W535–40.
71. Sehnal D, Bittrich S, Deshpande M, Svobodova R, Berka K, Bazgier V, et al. Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. Nucleic Acids Res. 2021;49(W1):W431–7.
72. Kauko A, Hedin LE, Thebaud E, Cristobal S, Elofsson A, von Heijne G. Repositioning of transmembrane alpha-helices during membrane protein folding. J Mol Biol. 2010;397(1):190–201.
73. Wang F, Cvirkaite-Krupovic V, Baquero DP, Krupovic M, Egelman EH. Cryo-EM of A. pernix flagellum.
74. Liu Y, Qi X, Li X. Catalytic and inhibitory mechanisms of porcupine-mediated Wnt acylation.
75. Xie T, Chi X, Huang B, Ye F, Zhou Q, Huang J. Rational exploration of fold atlas for human solute carrier proteins. Structure. 2022.
76. Farci D, Haniewicz P, de Sanctis D, Iesu L, Kereiche S, Winterhalter M, et al. The cryo-EM structure of the S-layer deinoxanthin-binding complex of Deinococcus radiodurans informs properties of its environmental interactions. J Biol Chem. 2022;298(6):102031.
77. Dolan KA, Kern DM, Kotecha A, Brohawn SG. Cryo-EM structure of SARS-CoV-2 M protein in lipid nanodiscs.
78. Pieper U, Schlessinger A, Kloppmann E, Chang GA, Chou JJ, Dumont ME, et al. Coordinating the impact of structural genomics on the human alpha-helical transmembrane proteome. Nat Struct Mol Biol. 2013;20(2):135–8.
79. Kloppmann E, Punta M, Rost B. Structural genomics plucks high-hanging membrane proteins. Curr Opin Struct Biol. 2012;22(3):326–32.

Publisher's Note

5. Conclusion

Looking back at the evolution of prediction methods for transmembrane proteins (TMP) over the last three decades, it is fascinating to see which parts changed and which stayed mostly the same (Table 1.1). For example, as newer and more sophisticated models became available, the field was quick to adapt; from simple threshold-based decisions, to statistical models, and finally complex machine learning models like neural networks (NN) and hidden Markov models (HMM). However, once HMMs and similar methods like conditional random fields (CRF) were established, they stayed for good. This makes sense given the well-structured "grammar" of TMPs. Similarly, most methods quickly adapted to using evolutionary information in one form or another as input, in order to improve their prediction performance. Another feature implemented by most modern methods is the inclusion of a signal peptide predictor, which helps to reduce false positive predictions. In contrast, the lack of prediction methods that include other membrane-embedded regions like re-entrant loops is still prevalent.

Although the significant gap of experimental 3D structures for TMP still exists, we now life in a time where the available tools and resources are able to close most of it. The gradual advances in sequence-based prediction methods for TMPs are closing in on the limit of what is possible with computational methods. Modern methods like TMbed [70] detect alpha-helical TMPs and beta-barrel TMPs with sensitivities in the mid to high 90s, while predicting less than one percent of soluble proteins as TMPs. Individual transmembrane segments are accurately predicted within only a few residues of their annotated locations nine out of ten times. As the error of such methods decreases, it is getting close to the error of the actual annotations in the available databases. For example, differences in the annotations between the Orientations of Proteins in Membranes [5] (OPM) database, the Protein Data Bank of Transmembrane Proteins [33–35] (PDBTM), and UniProtKB/Swiss-Prot [102] start to account for a significant portion

of the performance-differences in current evaluation tests [70]. Thus, one of the next efforts should be to focus on increasing the amount of high-quality annotations for known TMPs, which can then be used to train the next generation of machine learning methods.

Meanwhile, databases like TmAlphaFold [109] and TMvisDB [111] provide predicted 3D structures on a scale several magnitudes larger than the current number of experimentally determined 3D structures. Though the quality of predicted structures may vary, they are still based on the currently best-performing 3D structure prediction method, AlphaFold2 [87, 88]. They should be sufficient for many research projects where exact atomic coordinates are not required, or as starting points to select the best candidate proteins for experimental structure determination. Scanning those databases might also reveal novel folds of TMPs. For example, TMvisDB contains proteins with both predicted beta-barrel and alpha-helical transmembrane segments. Though some of those structures look plausible, there is currently no experimental evidence suggesting that such proteins exists. Further, advances in the computational generation of novel protein variants enable us to create thousands or even millions of new protein sequences in short amounts of time [123], which in turn can be scanned and filtered by the current generation of prediction methods. This truly opens up a completely new world of TMPs to discover and explore.

References

[1] von Heijne, G. The membrane protein universe: what's out there and why bother? *J Intern Med*, 261(6):543–57, 2007. 10.1111/j.1365-2796.2007.01792.x.

[2] Grouleff, J., Irudayam, S. J., Skeby, K. K., and Schiott, B. The influence of cholesterol on membrane protein structure, function, and dynamics studied by molecular dynamics simulations. *Biochim Biophys Acta*, 1848(9):1783–95, 2015. 10.1016/j.bbamem.2015.03.029.

[3] Enami, N., Okumua, H., and Kouyama, T. X-ray crystallographic study of archaerhodopsin. *Journal of Photoscience*, 9(2):320–322, 2002.

[4] Johansson, M. U., Alioth, S., Hu, K., Walser, R., Koebnik, R., et al. A minimal transmembrane beta-barrel platform protein studied by nuclear magnetic resonance. *Biochemistry*, 46(5):1128–40, 2007. 10.1021/bi061265e.

[5] Lomize, M. A., Pogozheva, I. D., Joo, H., Mosberg, H. I., and Lomize, A. L. Opm database and ppm web server: resources for positioning of proteins in membranes. *Nucleic Acids Res*, 40(Database issue):D370–6, 2012. 10.1093/nar/gkr703.

[6] Sehnal, D., Bittrich, S., Deshpande, M., Svobodova, R., Berka, K., et al. Mol* viewer: modern web app for 3d visualization and analysis of large biomolecular structures. *Nucleic Acids Res*, 49(W1):W431–W437, 2021. 10.1093/nar/gkab314.

[7] Fagerberg, L., Jonasson, K., von Heijne, G., Uhlen, M., and Berglund, L. Prediction of the human membrane proteome. *Proteomics*, 10(6):1141–9, 2010. 10.1002/pmic.200900258.

[8] Liu, J. and Rost, B. Comparing function and structure between entire proteomes. *Protein Sci*, 10(10):1970–9, 2001. 10.1110/ps.10101.

[9] Ulmschneider, M. B., Sansom, M. S., and Di Nola, A. Properties of integral membrane protein structures: derivation of an implicit membrane potential. *Proteins*, 59(2):252–65, 2005. 10.1002/prot.20334.

[10] Granseth, E., von Heijne, G., and Elofsson, A. A study of the membrane-water interface region of membrane proteins. *J Mol Biol*, 346(1):377–85, 2005. 10.1016/j.jmb.2004.11.036.

[11] Galdiero, S., Galdiero, M., and Pedone, C. beta-barrel membrane bacterial proteins: structure, function, assembly and interaction with lipids. *Curr Protein Pept Sci*, 8(1):63–82, 2007. 10.2174/138920307779941541.

[12] Garrow, A. G., Agnew, A., and Westhead, D. R. Tmb-hunt: an amino acid composition based method to screen proteomes for beta-barrel transmembrane proteins. *BMC Bioinformatics*, 6:56, 2005. 10.1186/1471-2105-6-56.

[13] Bigelow, H. R., Petrey, D. S., Liu, J., Przybylski, D., and Rost, B. Predicting transmembrane beta-barrels in proteomes. *Nucleic Acids Res*, 32(8):2566–77, 2004. 10.1093/nar/gkh580.

[14] de Planque, M. R., Bonev, B. B., Demmers, J. A., Greathouse, D. V., Koeppe, n., R. E., et al. Interfacial anchor properties of tryptophan residues in transmembrane peptides can dominate over hydrophobic matching effects in peptide-lipid interactions. *Biochemistry*, 42(18):5341–8, 2003. 10.1021/bi027000r.

[15] Yau, W. M., Wimley, W. C., Gawrisch, K., and White, S. H. The preference of tryptophan for membrane interfaces. *Biochemistry*, 37(42):14713–8, 1998. 10.1021/bi980809c.

[16] Sun, H., Greathouse, D. V., Andersen, O. S., and Koeppe, n., R. E. The preference of tryptophan for membrane interfaces: insights from n-methylation of tryptophans in gramicidin channels. *J Biol Chem*, 283(32):22233–43, 2008. 10.1074/jbc.M802074200.

[17] Heijne, G. The distribution of positively charged residues in bacterial inner membrane proteins correlates with the trans-membrane topology. *EMBO J*, 5(11):3021–7, 1986. 10.1002/j.1460-2075.1986.tb04601.x.

[18] von Heijne, G. and Gavel, Y. Topogenic signals in integral membrane proteins. *Eur J Biochem*, 174(4):671–8, 1988. 10.1111/j.1432-1033.1988.tb14150.x.

[19] von Heijne, G. Analysis of the distribution of charged residues in the n-terminal region of signal sequences: implications for protein export in prokaryotic and eukaryotic cells. *EMBO J*, 3(10):2315–8, 1984. 10.1002/j.1460-2075.1984.tb02132.x.

[20] Nilsson, I. and von Heijne, G. Fine-tuning the topology of a polytopic membrane protein: role of positively and negatively charged amino acids. *Cell*, 62(6):1135–41, 1990. 10.1016/0092-8674(90)90390-z.

[21] von Heijne, G. Control of topology and mode of assembly of a polytopic membrane protein by positively charged residues. *Nature*, 341(6241):456–8, 1989. 10.1038/341456a0.

[22] Overington, J. P., Al-Lazikani, B., and Hopkins, A. L. How many drug targets are there? *Nat Rev Drug Discov*, 5(12):993–6, 2006. 10.1038/nrd2199.

[23] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., et al. The protein data bank. *Nucleic Acids Res*, 28(1):235–42, 2000. 10.1093/nar/28.1.235.

[24] Hendrickson, W. A. Atomic-level analysis of membrane-protein structure. *Nat Struct Mol Biol*, 23(6):464–7, 2016. 10.1038/nsmb.3215.

[25] Varga, J., Dobson, L., Remenyi, I., and Tusnady, G. E. Tstmp: target selection for structural genomics of human transmembrane proteins. *Nucleic Acids Res*, 45(D1):D325–D330, 2017. 10.1093/nar/gkw939.

[26] Newport, T. D., Sansom, M. S. P., and Stansfeld, P. J. The memprotmd database: a resource for membrane-embedded protein structures and their lipid interactions. *Nucleic Acids Res*, 47(D1):D390–D397, 2019. 10.1093/nar/gky1047.

[27] Carpenter, E. P., Beis, K., Cameron, A. D., and Iwata, S. Overcoming the challenges of membrane protein crystallography. *Curr Opin Struct Biol*, 18(5):581–6, 2008. 10.1016/j.sbi.2008.07.001.

[28] Drew, D., Froderberg, L., Baars, L., and de Gier, J. W. Assembly and overexpression of membrane proteins in escherichia coli. *Biochim Biophys Acta*, 1610(1):3–10, 2003. 10.1016/s0005-2736(02)00707-1.

[29] Hunte, C., Koepke, J., Lange, C., Rossmanith, T., and Michel, H. Structure at 2.3 a resolution of the cytochrome bc(1) complex from the yeast saccharomyces cerevisiae co-crystallized with an antibody fv fragment. *Structure*, 8(6):669–84, 2000. 10.1016/s0969-2126(00)00152-0.

[30] Cherezov, V., Rosenbaum, D. M., Hanson, M. A., Rasmussen, S. G., Thian, F. S., et al. High-resolution crystal structure of an engineered human beta2-adrenergic g protein-coupled receptor. *Science*, 318(5854):1258–65, 2007. 10.1126/science.1150577.

[31] Rosenbaum, D. M., Cherezov, V., Hanson, M. A., Rasmussen, S. G., Thian, F. S., et al. Gpcr engineering yields high-resolution structural insights into beta2-adrenergic receptor function. *Science*, 318(5854):1266–73, 2007. 10.1126/science.1150609.

[32] Milne, J. L., Borgnia, M. J., Bartesaghi, A., Tran, E. E., Earl, L. A., et al. Cryo-electron microscopy–a primer for the non-microscopist. *FEBS J*, 280(1):28–45, 2013. 10.1111/febs.12078.

[33] Tusnady, G. E., Dosztanyi, Z., and Simon, I. Transmembrane proteins in the protein data bank: identification and classification. *Bioinformatics*, 20(17):2964–72, 2004. 10.1093/bioinformatics/bth340.

[34] Tusnady, G. E., Dosztanyi, Z., and Simon, I. Pdb_tm: selection and membrane localization of transmembrane proteins in the protein data bank. *Nucleic Acids Res*, 33(Database issue):D275–8, 2005. 10.1093/nar/gki002.

[35] Kozma, D., Simon, I., and Tusnady, G. E. Pdbtm: Protein data bank of transmembrane proteins after 8 years. *Nucleic Acids Res*, 41(Database issue):D524–9, 2013. 10.1093/nar/gks1169.

[36] Lomize, A. L., Pogozheva, I. D., Lomize, M. A., and Mosberg, H. I. Positioning of proteins in membranes: a computational approach. *Protein Sci*, 15(6):1318–33, 2006. 10.1110/ps.062126106.

[37] Lomize, A. L., Pogozheva, I. D., and Mosberg, H. I. Anisotropic solvent model of the lipid bilayer. 2. energetics of insertion of small molecules, peptides, and proteins in membranes. *J Chem Inf Model*, 51(4):930–46, 2011. 10.1021/ci200020k.

[38] Lomize, A. L., Todd, S. C., and Pogozheva, I. D. Spatial arrangement of proteins in planar and curved membranes by ppm 3.0. *Protein Sci*, 31(1):209–220, 2022. 10.1002/pro.4219.

[39] Tusnady, G. E., Dosztanyi, Z., and Simon, I. Tmdet: web server for detecting transmembrane regions of proteins by using their 3d coordinates. *Bioinformatics*, 21(7):1276–7, 2005. 10.1093/bioinformatics/bti121.

[40] von Heijne, G. Membrane protein structure prediction. hydrophobicity analysis and the positive-inside rule. *J Mol Biol*, 225(2):487–94, 1992. 10.1016/0022-2836(92)90934-c.

[41] Jones, D. T., Taylor, W. R., and Thornton, J. M. A model recognition approach to the prediction of all-helical membrane protein structure and topology. *Biochemistry*, 33(10):3038–49, 1994. 10.1021/bi00176a037.

[42] Rost, B., Casadio, R., Fariselli, P., and Sander, C. Transmembrane helices predicted at 95*Protein Sci*, 4(3):521–33, 1995. 10.1002/pro.5560040318.

[43] Rost, B., Fariselli, P., and Casadio, R. Topology prediction for helical transmembrane proteins at 86accuracy. *Protein Sci*, 5(8):1704–18, 1996. 10.1002/pro.5560050824.

[44] Rost, B., Casadio, R., and Fariselli, P. Refining neural network predictions for helical transmembrane proteins by dynamic programming. *Proc Int Conf Intell Syst Mol Biol*, 4:192–200, 1996.

[45] Claros, M. G. and von Heijne, G. Toppred ii: an improved software for membrane protein structure predictions. *Comput Appl Biosci*, 10(6):685–6, 1994. 10.1093/bioinformatics/10.6.685.

[46] Jones, D. T. Do transmembrane protein superfolds exist? *FEBS Lett*, 423(3):281–5, 1998. 10.1016/s0014-5793(98)00095-7.

[47] Sonnhammer, E. L., von Heijne, G., and Krogh, A. A hidden markov model for predicting transmembrane helices in protein sequences. *Proc Int Conf Intell Syst Mol Biol*, 6:175–82, 1998.

[48] Krogh, A., Larsson, B., von Heijne, G., and Sonnhammer, E. L. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *J Mol Biol*, 305(3):567–80, 2001. 10.1006/jmbi.2000.4315.

[49] Tusnady, G. E. and Simon, I. Principles governing amino acid composition of integral membrane proteins: application to topology prediction. *J Mol Biol*, 283(2):489–506, 1998. 10.1006/jmbi.1998.2107.

[50] Tusnady, G. E. and Simon, I. The hmmtop transmembrane topology prediction server. *Bioinformatics*, 17(9):849–50, 2001. 10.1093/bioinformatics/17.9.849.

[51] Hirokawa, T., Boon-Chieng, S., and Mitaku, S. Sosui: classification and secondary structure prediction system for membrane proteins. *Bioinformatics*, 14(4):378–9, 1998. 10.1093/bioinformatics/14.4.378.

[52] Kall, L., Krogh, A., and Sonnhammer, E. L. A combined transmembrane topology and signal peptide prediction method. *J Mol Biol*, 338(5):1027–36, 2004. 10.1016/j.jmb.2004.03.016.

[53] Bigelow, H. and Rost, B. Proftmb: a web server for predicting bacterial transmembrane beta barrel proteins. *Nucleic Acids Res*, 34(Web Server issue):W186–8, 2006. 10.1093/nar/gkl262.

[54] Kall, L., Krogh, A., and Sonnhammer, E. L. An hmm posterior decoder for sequence feature prediction that includes homology information. *Bioinformatics*, 21 Suppl 1:i251–7, 2005. 10.1093/bioinformatics/bti1014.

[55] Jones, D. T. Improving the accuracy of transmembrane protein topology prediction using evolutionary information. *Bioinformatics*, 23(5):538–44, 2007. 10.1093/bioinformatics/btl677.

[56] Viklund, H. and Elofsson, A. Octopus: improving topology prediction by two-track ann-based preference scores and an extended topological grammar. *Bioinformatics*, 24(15):1662–8, 2008. 10.1093/bioinformatics/btn221.

[57] Viklund, H., Bernsel, A., Skwark, M., and Elofsson, A. Spoctopus: a combined predictor of signal peptides and membrane protein topology. *Bioinformatics*, 24(24):2928–9, 2008. 10.1093/bioinformatics/btn550.

[58] Nugent, T. and Jones, D. T. Transmembrane protein topology prediction using support vector machines. *BMC Bioinformatics*, 10:159, 2009. 10.1186/1471-2105-10-159.

[59] Bernsel, A., Viklund, H., Hennerdal, A., and Elofsson, A. Topcons: consensus prediction of membrane protein topology. *Nucleic Acids Res*, 37(Web Server issue):W465–8, 2009. 10.1093/nar/gkp363.

[60] Hayat, S. and Elofsson, A. Boctopus: improved topology prediction of transmembrane beta barrel proteins. *Bioinformatics*, 28(4):516–22, 2012. 10.1093/bioinformatics/btr710.

[61] Savojardo, C., Fariselli, P., and Casadio, R. Betaware: a machine-learning tool to detect and predict transmembrane beta-barrel proteins in prokaryotes. *Bioinformatics*, 29(4):504–5, 2013. 10.1093/bioinformatics/bts728.

[62] Tsirigos, K. D., Peters, C., Shu, N., Kall, L., and Elofsson, A. The topcons web server for consensus prediction of membrane protein topology and signal peptides. *Nucleic Acids Res*, 43(W1):W401–7, 2015. 10.1093/nar/gkv485.

[63] Dobson, L., Remenyi, I., and Tusnady, G. E. Cctop: a consensus constrained topology prediction web server. *Nucleic Acids Res*, 43(W1):W408–12, 2015. 10.1093/nar/gkv451.

[64] Dobson, L., Remenyi, I., and Tusnady, G. E. The human transmembrane proteome. *Biol Direct*, 10:31, 2015. 10.1186/s13062-015-0061-x.

[65] Bernhofer, M., Kloppmann, E., Reeb, J., and Rost, B. Tmseg: Novel prediction of transmembrane helices. *Proteins*, 84(11):1706–1716, 2016. 10.1002/prot.25155.

[66] Hayat, S., Peters, C., Shu, N., Tsirigos, K. D., and Elofsson, A. Inclusion of dyad-repeat pattern improves topology prediction of transmembrane beta-barrel proteins. *Bioinformatics*, 32(10):1571–3, 2016. 10.1093/bioinformatics/btw025.

[67] Madeo, G., Savojardo, C., Martelli, P. L., and Casadio, R. Betaware-deep: An accurate web server for discrimination and topology prediction of prokaryotic transmembrane beta-barrel proteins. *J Mol Biol*, 433(11):166729, 2021. 10.1016/j.jmb.2020.166729.

[68] Hallgren, J., Tsirigos, K. D., Pedersen, M. D., Almagro Armenteros, J. J., Marcatili, P., et al. Deeptmhmm predicts alpha and beta transmembrane proteins using deep neural networks. *bioRxiv*, page 2022.04.08.487609, 2022. 10.1101/2022.04.08.487609.

[69] Wang, L., Zhong, H., Xue, Z., and Wang, Y. Improving the topology prediction of alpha-helical transmembrane proteins with deep transfer learning. *Comput Struct Biotechnol J*, 20:1993–2000, 2022. 10.1016/j.csbj.2022.04.024.

[70] Bernhofer, M. and Rost, B. Tmbed: transmembrane proteins predicted through language model embeddings. *BMC Bioinformatics*, 23(1):326, 2022. 10.1186/s12859-022-04873-x.

[71] Henikoff, S. and Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, 89(22):10915–9, 1992. 10.1073/pnas.89.22.10915.

[72] Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., et al. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–402, 1997. 10.1093/nar/25.17.3389.

[73] Remmert, M., Biegert, A., Hauser, A., and Soding, J. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nat Methods*, 9(2):173–5, 2011. 10.1038/nmeth.1818.

[74] Mirdita, M., Steinegger, M., and Soding, J. Mmseqs2 desktop and local web server app for fast, interactive sequence searches. *Bioinformatics*, 35(16):2856–2858, 2019. 10.1093/bioinformatics/bty1057.

[75] Steinegger, M. and Soding, J. Clustering huge protein sequence sets in linear time. *Nat Commun*, 9(1):2542, 2018. 10.1038/s41467-018-04964-5.

[76] Steinegger, M. and Soding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol*, 35(11):1026–1028, 2017. 10.1038/nbt.3988.

[77] Gribskov, M., McLachlan, A. D., and Eisenberg, D. Profile analysis: detection of distantly related proteins. *Proc Natl Acad Sci U S A*, 84(13):4355–8, 1987. 10.1073/pnas.84.13.4355.

[78] Nielsen, H. and Krogh, A. Prediction of signal peptides and signal anchors by a hidden markov model. *Proc Int Conf Intell Syst Mol Biol*, 6:122–30, 1998.

[79] Bernsel, A., Viklund, H., Falk, J., Lindahl, E., von Heijne, G., et al. Prediction of membrane-protein topology from first principles. *Proc Natl Acad Sci U S A*, 105(20):7177–81, 2008. 10.1073/pnas.0711151105.

[80] Viklund, H. and Elofsson, A. Best alpha-helical transmembrane protein topology predictions are achieved using hidden markov models and evolutionary information. *Protein Sci*, 13(7):1908–17, 2004. 10.1110/ps.04625404.

[81] Reynolds, S. M., Kall, L., Riffle, M. E., Bilmes, J. A., and Noble, W. S. Transmembrane topology and signal peptide prediction using dynamic bayesian networks. *PLoS Comput Biol*, 4(11):e1000213, 2008. 10.1371/journal.pcbi.1000213.

[82] Shen, H. and Chou, J. J. Membrain: improving the accuracy of predicting transmembrane helices. *PLoS One*, 3(6):e2399, 2008. 10.1371/journal.pone.0002399.

[83] Fariselli, P., Savojardo, C., Martelli, P. L., and Casadio, R. Grammatical-restrained hidden conditional random fields for bioinformatics applications. *Algorithms Mol Biol*, 4:13, 2009. 10.1186/1748-7188-4-13.

[84] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., et al. Attention is all you need. 2017.

[85] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2020. 10.48550/arXiv.2010.11929.

[86] Verma, P. and Berger, J. Audio transformers:transformer architectures for large scale audio understanding. adieu convolutions. 2021. 10.48550/arXiv.2105.00335.

[87] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. 10.1038/s41586-021-03819-2.

[88] Marx, V. Method of the year: protein structure prediction. *Nat Methods*, 19(1):5–10, 2022. 10.1038/s41592-021-01359-1.

[89] Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., et al. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Trans Pattern Anal Mach Intell*, PP, 2021. 10.1109/TPAMI.2021.3095381.

[90] Bepler, T. and Berger, B. Learning the protein language: Evolution, structure, and function. *Cell Syst*, 12(6):654–669 e3, 2021. 10.1016/j.cels.2021.05.017.

[91] Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci U S A*, 118(15), 2021. 10.1073/pnas.2016239118.

[92] Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., et al. Language models enable zero-shot prediction of the effects of mutations on protein function. 2021.

[93] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., et al. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, page 2022.07.20.500902, 2022. 10.1101/2022.07.20.500902.

[94] Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. 2021. 10.48550/arXiv.2108.12409.

[95] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., et al. Exploring the limits of transfer learning with a unified text-to-text transformer. 2019. 10.48550/arXiv.1910.10683.

[96] Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., et al. Roformer: Enhanced transformer with rotary position embedding. 2021. 10.48550/arXiv.2104.09864.

[97] Devlin, J., Chang, M., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018. 10.48550/arXiv.1810.04805.

[98] Alec, R., Karthik, N., Tim, S., and Ilya, S. Improving language understanding by generative pre-training. 2018.

[99] Alec, R., Jeffrey, W., Rewon, C., David, L., Dario, A., et al. Language models are unsupervised multitask learners. 2019.

[100] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., et al. Language models are few-shot learners. 2020. 10.48550/arXiv.2005.14165.

[101] Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., et al. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, 20(1):723, 2019. 10.1186/s12859-019-3220-8.

[102] UniProt, C. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Res*, 51(D1):D523–D531, 2023. 10.1093/nar/gkac1052.

[103] Steinegger, M., Mirdita, M., and Soding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat Methods*, 16(7):603–606, 2019. 10.1038/s41592-019-0437-4.

[104] Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T., and Rost, B. Embeddings from deep learning transfer go annotations beyond homology. *Sci Rep*, 11(1):1160, 2021. 10.1038/s41598-020-80786-0.

[105] Marquet, C., Heinzinger, M., Olenyi, T., Dallago, C., Erckert, K., et al. Embeddings from protein language models predict conservation and variant effects. *Hum Genet*, 2021. 10.1007/s00439-021-02411-y.

[106] Littmann, M., Heinzinger, M., Dallago, C., Weissenow, K., and Rost, B. Protein embeddings and deep learning predict binding residues for various ligand classes. *Sci Rep*, 11(1):23916, 2021. 10.1038/s41598-021-03431-4.

[107] Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res*, 50(D1):D439–D444, 2022. 10.1093/nar/gkab1061.

[108] Hegedus, T., Geisler, M., Lukacs, G. L., and Farkas, B. Ins and outs of alphafold2 transmembrane protein structure predictions. *Cell Mol Life Sci*, 79(1):73, 2022. 10.1007/s00018-021-04112-1.

[109] Dobson, L., Szekeres, L. I., Gerdan, C., Lango, T., Zeke, A., et al. Tmalphafold database: membrane localization and evaluation of alphafold2 predicted alpha-helical transmembrane protein structures. *Nucleic Acids Res*, 51(D1):D517–D522, 2023. 10.1093/nar/gkac928.

[110] Teufel, F., Almagro Armenteros, J. J., Johansen, A. R., Gislason, M. H., Pihl, S. I., et al. Signalp 6.0 predicts all five types of signal peptides using protein language models. *Nat Biotechnol*, 40(7):1023–1025, 2022. 10.1038/s41587-021-01156-3.

[111] Marquet, C., Grekova, A., Houri, L., Bernhofer, M., Jimenez-Soto, L. F., et al. Tmvisdb: resource for transmembrane protein annotation and 3d visualization. *bioRxiv*, page 2022.11.30.518551, 2022. 10.1101/2022.11.30.518551.

[112] Bernhofer, M., Dallago, C., Karl, T., Satagopam, V., Heinzinger, M., et al. Predictprotein - predicting protein structure and function for 29 years. *Nucleic Acids Res*, 49(W1):W535–W540, 2021. 10.1093/nar/gkab354.

[113] Yachdav, G., Kloppmann, E., Kajan, L., Hecht, M., Goldberg, T., et al. Predictprotein–an open resource for online prediction of protein structural and functional features. *Nucleic Acids Res*, 42(Web Server issue):W337–43, 2014. 10.1093/nar/gku366.

[114] Goldberg, T., Hecht, M., Hamp, T., Karl, T., Yachdav, G., et al. Loctree3 prediction of localization. *Nucleic Acids Res*, 42(Web Server issue):W350–5, 2014. 10.1093/nar/gku396.

[115] Hecht, M., Bromberg, Y., and Rost, B. Better prediction of functional effects for sequence variants. *BMC Genomics*, 16 Suppl 8(Suppl 8):S1, 2015. 10.1186/1471-2164-16-S8-S1.

[116] Petersen, T. N., Brunak, S., von Heijne, G., and Nielsen, H. Signalp 4.0: discriminating signal peptides from transmembrane regions. *Nat Methods*, 8(10):785–6, 2011. 10.1038/nmeth.1701.

[117] Watkins, X., Garcia, L. J., Pundir, S., Martin, M. J., and UniProt, C. Protvista: visualization of protein sequence annotations. *Bioinformatics*, 33(13):2040–2041, 2017. 10.1093/bioinformatics/btx120.

[118] Reguant, R., Antipin, Y., Sheridan, R., Dallago, C., Diamantoukos, D., et al. Alignmentviewer: Sequence analysis of large protein families. *F1000Res*, 9:epublish, 2020. 10.12688/f1000research.22242.2.

[119] Dallago, C., Goldberg, T., Andrade-Navarro, M. A., Alanis-Lobato, G., and Rost, B. Visualizing human protein-protein interactions and subcellular localizations on cell images through cellmap. *Curr Protoc Bioinformatics*, 69(1):e97, 2020. 10.1002/cpbi.97.

[120] Qiu, J., Bernhofer, M., Heinzinger, M., Kemper, S., Norambuena, T., et al. Prona2020 predicts protein-dna, protein-rna, and protein-protein binding proteins and residues from sequence. *J Mol Biol*, 432(7):2428–2443, 2020. 10.1016/j.jmb.2020.02.026.

[121] Reeb, J., Kloppmann, E., Bernhofer, M., and Rost, B. Evaluation of transmembrane helix predictions in 2014. *Proteins*, 83(3):473–84, 2015. 10.1002/prot.24749.

[122] Olenyi, T., Marquet, C., Heinzinger, M., Kroger, B., Nikolova, T., et al. Lambdapp: Fast and accessible protein-specific phenotype predictions. *Protein Sci*, 32(1):e4524, 2023. 10.1002/pro.4524.

[123] Ferruz, N., Schmidt, S., and Hocker, B. Protgpt2 is a deep unsupervised language model for protein design. *Nat Commun*, 13(1):4348, 2022. 10.1038/s41467-022-32007-7.

A. Appendix

List of Publications

The publication-based dissertation at hand is based on the following 3 peer-reviewed and published publications:

- Bernhofer, M., Kloppmann, E., Reeb, J., and Rost, B. Tmseg: Novel prediction of transmembrane helices. *Proteins*, 84(11):1706–1716, 2016. 10.1002/prot.25155
- Bernhofer, M., Dallago, C., Karl, T., Satagopam, V., Heinzinger, M., et al. Predictprotein - predicting protein structure and function for 29 years. *Nucleic Acids Res*, 49(W1):W535–W540, 2021. 10.1093/nar/gkab354
- Bernhofer, M. and Rost, B. Tmbed: transmembrane proteins predicted through language model embeddings. *BMC Bioinformatics*, 23(1):326, 2022. 10.1186/s12859-022-04873-x

In addition to the publications above, I have co-authored the following publications cited in this dissertation:

- Goldberg, T., Hecht, M., Hamp, T., Karl, T., Yachdav, G., et al. Loctree3 prediction of localization. *Nucleic Acids Res*, 42(Web Server issue):W350–5, 2014. 10.1093/nar/gku396
- Yachdav, G., Kloppmann, E., Kajan, L., Hecht, M., Goldberg, T., et al. Predictprotein–an open resource for online prediction of protein structural and functional features. *Nucleic Acids Res*, 42(Web Server issue):W337–43, 2014. 10.1093/nar/gku366
- Reeb, J., Kloppmann, E., Bernhofer, M., and Rost, B. Evaluation of transmembrane helix predictions in 2014. *Proteins*, 83(3):473–84, 2015. 10.1002/prot.24749
- Qiu, J., Bernhofer, M., Heinzinger, M., Kemper, S., Norambuena, T., et al. Prona2020 predicts protein-dna, protein-rna, and protein-protein binding proteins and residues from sequence. *J Mol Biol*, 432(7):2428–2443, 2020. 10.1016/j.jmb.2020.02.026

- Marquet, C., Heinzinger, M., Olenyi, T., Dallago, C., Erckert, K., et al. Embeddings from protein language models predict conservation and variant effects. *Hum Genet*, 2021. 10.1007/s00439-021-02411-y

- Olenyi, T., Marquet, C., Heinzinger, M., Kroger, B., Nikolova, T., et al. Lambdapp: Fast and accessible protein-specific phenotype predictions. *Protein Sci*, 32(1):e4524, 2023. 10.1002/pro.4524

Further, I have co-authored the following publication that has been submitted to peer-review and has been published as pre-print on bioRxiv. The results are cited in this dissertation.

- Marquet, C., Grekova, A., Houri, L., Bernhofer, M., Jimenez-Soto, L. F., et al. Tmvisdb: resource for transmembrane protein annotation and 3d visualization. *bioRxiv*, page 2022.11.30.518551, 2022. 10.1101/2022.11.30.518551

Finally, I have co-authored the following publications not discussed in this dissertation:

- Hucker, S. M., Ardern, Z., Goldberg, T., Schafferhans, A., Bernhofer, M., et al. Discovery of numerous novel small genes in the intergenic regions of the escherichia coli o157:h7 sakai genome. *PLoS One*, 12(9):e0184119, 2017. 10.1371/journal.pone.0184119

- Bernhofer, M., Goldberg, T., Wolf, S., Ahmed, M., Zaugg, J., et al. Nlsdb-major update for database of nuclear localization signals and nuclear export signals. *Nucleic Acids Res*, 46(D1):D503–D508, 2018. 10.1093/nar/gkx1021

- Marot-Lassauzaie, V., Bernhofer, M., and Rost, B. Correcting mistakes in predicting distributions. *Bioinformatics*, 34(19):3385–3386, 2018. 10.1093/bioinformatics/bty346

- Peeken, J. C., Bernhofer, M., Wiestler, B., Goldberg, T., Cremers, D., et al. Radiomics in radiooncology - challenging the medical physicist. *Phys Med*, 48:27–36, 2018. 10.1016/j.ejmp.2018.03.012

- Peeken, J. C., Goldberg, T., Knie, C., Komboz, B., Bernhofer, M., et al. Treatment-related features improve machine learning prediction of prognosis in soft tissue sarcoma patients. *Strahlenther Onkol*, 194(9):824–834, 2018. 10.1007/s00066-018-1294-2

- Peeken, J. C., Bernhofer, M., Spraker, M. B., Pfeiffer, D., Devecka, M., et al. Ct-based radiomic features predict tumor grading and have prognostic value in patients with soft tissue sarcomas treated with neoadjuvant radiation therapy. *Radiother Oncol*, 135:187–196, 2019. 10.1016/j.radonc.2019.01.004

- Peeken, J. C., Goldberg, T., Pyka, T., Bernhofer, M., Wiestler, B., et al. Combining multimodal imaging and treatment features improves machine learningbased prognostic assessment in patients with glioblastoma multiforme. *Cancer Med*, 8(1):128–136, 2019. 10.1002/cam4.1908

A.1. Publications Included in This Dissertation

A.1.1. Journal Article: Michael Bernhofer *et al.*, Proteins (2016)

PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

TMSEG: Novel prediction of transmembrane helices

Michael Bernhofer,[1]* Edda Kloppmann,[1,2] Jonas Reeb,[1] and Burkhard Rost[1,2,3,4]

[1] Department of Informatics & Center for Bioinformatics & Computational Biology – i12, Technische Universität München (TUM), Boltzmannstr. 3, Garching/Munich 85748, Germany

[2] New York Consortium on Membrane Protein Structure, New York Structural Biology Center, New York, New York 10027

[3] Institute of Advanced Study (TUM-IAS), Lichtenbergstr. 2a, Garching/Munich 85748, Germany

[4] Institute for Food and Plant Sciences WZW – Weihenstephan, Alte Akademie 8, Freising, Germany

ABSTRACT

Transmembrane proteins (TMPs) are important drug targets because they are essential for signaling, regulation, and transport. Despite important breakthroughs, experimental structure determination remains challenging for TMPs. Various methods have bridged the gap by predicting transmembrane helices (TMHs), but room for improvement remains. Here, we present TMSEG, a novel method identifying TMPs and accurately predicting their TMHs and their topology. The method combines machine learning with empirical filters. Testing it on a non-redundant dataset of 41 TMPs and 285 soluble proteins, and applying strict performance measures, TMSEG outperformed the state-of-the-art in our hands. TMSEG correctly distinguished helical TMPs from other proteins with a sensitivity of 98 ± 2% and a false positive rate as low as 3 ± 1%. Individual TMHs were predicted with a precision of 87 ± 3% and recall of 84 ± 3%. Furthermore, in 63 ± 6% of helical TMPs the placement of all TMHs and their inside/outside topology was correctly predicted. There are two main features that distinguish TMSEG from other methods. First, the errors in finding all helical TMPs in an organism are significantly reduced. For example, in human this leads to 200 and 1600 fewer misclassifications compared to the second and third best method available, and 4400 fewer mistakes than by a simple hydrophobicity-based method. Second, TMSEG provides an add-on improvement for any existing method to benefit from.

Proteins 2016; 84:1706–1716.

Key words: membrane protein; protein structure prediction; transmembrane helices; α-helical integral membrane protein; transmembrane protein prediction; transmembrane helix prediction.

INTRODUCTION

Transmembrane proteins (TMPs) are involved in numerous essential processes within living organisms such as signaling, regulation, and transport.[1] About 20–30% of all proteins within any organism have been estimated to be TMPs.[2,3] Many TMPs, especially G protein-coupled receptors (GPCRs), are primary drug targets[4] and therefore of high interest.

TMPs cross the membrane bilayer with either transmembrane helices (TMHs) or beta-strands. The latter are found in the outer membrane of Gram-negative bacteria, mitochondria, and chloroplasts. They make up only about 1–2% of all proteins in Gram-negative bacteria.[5] We concentrated on the more common class of helical TMPs and will refer to these as TMPs in the following. TMPs can cross the membrane only once (single-pass) or multiple times (multi-pass). Due to the apolar and hydrophobic environment in the lipid bilayer, most of the amino acids found in TMHs are hydrophobic, and their orientation in the membrane (called TMP topology) can be discerned through Gunnar von Heijne's positive-inside rule.[6,7]

Additional Supporting Information may be found in the online version of this article.

Abbreviations used: 3D, three-dimensional; GPCR, G protein-coupled receptor; NN, (artificial) neural network; OPM, Orientations of Proteins in Membranes; PDB, Protein Data Bank; PDBTM, Protein Data Bank of Transmembrane Proteins; RF, random forest; TMH, transmembrane alpha-helix; TMP, transmembrane protein.

Grant sponsor: Alexander von Humboldt Foundation; Grant sponsor: National Institutes of Health (NIH); Grant number: U54 GM095315.
*Correspondence to: Michael.Bernhofer@mytum.de
Received 22 May 2016; Revised 18 July 2016; Accepted 24 August 2016
Published online 26 August 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/prot.25155

Despite their immense importance, and despite crucial experimental advances,[8–11] <2% of the structures in the Protein Data Bank[12] (PDB) are TMPs.[13–15] As membrane regions are typically not visible in high-resolution structures, TMHs are assigned to PDB structures by expert resources, most prominently the Orientations of Proteins in Membranes[16] (OPM) database and the Protein Data Bank of Transmembrane Proteins[17] (PDBTM).

Recent advances in experimental structure determination have benefited from advanced computational predictions of TMHs from sequence.[8,9] In the last 25 years, many such tools have been developed, ranging from simple algorithms based solely on hydrophobicity scales (e.g., TopPred[18]) to sophisticated uses of hidden Markov models (e.g., TMHMM,[19] HMMTOP,[20] Phobius,[21] and PolyPhobius[22]), neural networks (e.g., PHDhtm,[23,24] and MEMSAT3[25]), and support vector machines (MEMSAT-SVM[26]). Arguably, the most important advance was the incorporation of evolutionary information from sequence profiles or multiple sequence alignments.[23,24] Consequently, almost all methods developed over the last decade are based on evolutionary information. A recent assessment applying strict evaluation measures showed that many methods perform well overall; the best are some recent methods.[27] Here, we show that a few simple ideas improve significantly over the state-of-the-art.

MATERIAL AND METHODS

Dataset TMP166: helical TMPs with known structures

We collected helical TMPs with known structures annotated in OPM[16] and PDBTM[17] (releases 2013_07). Both databases use PDB[12] chain identifiers. We mapped those PDB chains to their UniProtKB[28] protein sequences using SIFTS.[29] We excluded all chimeric PDB chains, model structures, X-ray structures with >8 Å, and those for which some TMH residues did not map gapless to UniProtKB sequences. This gave 1087 PDB chains from 455 PDB structures (379 X-ray and 76 NMR structures).

UniqueProt[30] reduced sequence-redundancy at HVAL > 0 (the HVAL depends on alignment length and the percentage of pairwise sequence identity[31]). At this threshold, no pair of proteins has >20% pairwise sequence identity for alignments of >250 residues (see Rost 1999[32] for precise definitions). The result of this is our final dataset consisting of 166 non-redundant TMPs (called TMP166, Supporting Information Table S1).

As the TMH annotations in OPM and PDBTM differed for some proteins, we associated TMH annotations from both databases with each sequence. The inside/outside topology of the non-transmembrane regions was assigned based on the ATOM coordinates and topology annotation from OPM (cf. Note Supporting Information S1 and Fig. S1). We considered re-entrant regions[33,34] to be non-transmembrane due to their scarcity in the TMP166 dataset (only 15 proteins with one or two re-entry regions each; Supporting Information Table S1).

Dataset SP1441: proteins with and without signal peptides

As signal peptides are often confused with TMHs and vice versa,[27] a second dataset was derived from the SignalP4.1 dataset.[35] This dataset contained UniProtKB sequences of soluble proteins and TMPs with and without signal peptide annotations. Note that these TMPs have no inside/outside topology annotations and many of their TMH annotations are not supported by experimental evidence.

The SignalP4.1 dataset was redundancy reduced twice using UniqueProt. First, all proteins similar to any of those in the TMP166 dataset were removed at HVAL > 0. Second, the remaining proteins were redundancy-filtered at HVAL > 0. The final dataset contained 1441 proteins sequences (299 TMPs and 1142 soluble proteins, called SP1441; Supporting Information Table S2). About 477 of those had signal peptide annotations (25 TMPs and 452 soluble proteins).

Splitting the datasets

We split the combined TMP166 and SP1441 dataset into four subsets. We partitioned them in a way that all subsets have approximately the same distributions with respect to the number of soluble proteins and TMPs, protein sequences with and without signal peptides, and sequence lengths (Supporting Information Fig. S2).

We used the first three subsets to develop TMSEG in a three-fold cross-validation approach (cf. TMSEG training). The fourth split, the independent test set called BlindTest, was used only for the final performance evaluation, i.e., no parameter was optimized on that set. The BlindTest dataset contained 41 TMPs (from TMP166) with known structure and TMH annotations from OPM and PDBTM, and 285 soluble proteins from the SP1441 dataset. The 74 TMPs from the fourth split of SP1441 (Supporting Information Table S2) were not included in the BlindTest dataset, because they lack sufficient experimental annotations. However, we used them for the signal peptide prediction performance analysis, as we did not have curated signal peptide annotations for the TMPs from OPM and PDBTM.

Human proteome

We retrieved the human proteome, 20,196 protein sequences, from UniProtKB/Swiss-Prot (release 2015_03). We applied our TMSEG algorithm to the whole proteome to provide a summary of its TMP composition and to estimate run time.

Table I
Evaluation Measures

Measurement	Formula	Description
Precision (%)	$100 * \frac{\#\text{ of correctly predicted TMHs}}{\#\text{ of predicted TMHs}}$	Precision of TMH prediction
Recall (%)	$100 * \frac{\#\text{ of correctly predicted TMHs}}{\#\text{ of observed TMHs}}$	Recall of TMH prediction
Q_{ok} (%)	$\frac{100}{N} * \sum_{i=1}^{N} x_i;\ x_i = \begin{cases} 1, & if\ p_i = r_i = 100\% \\ 0, & else \end{cases}$	Percentage of TMPs with correct TMH placement
Q_{top} (%)	$\frac{100}{N} * \sum_{i=1}^{N} y_i;\ y_i = \begin{cases} 1, & if\ p_i = r_i = t_i = 100\% \\ 0, & else \end{cases}$	Percentage of TMPs with correct TMH placement and inside/outside topology
FPR (%)	$100 * \frac{\#\text{ of incorrectly predicted TMPs}}{\#\text{ of soluble proteins}}$	False positive rate of TMP prediction
Sensitivity (%)	$100 * \frac{\#\text{ of correctly predicted TMPs}}{\#\text{ of observed TMPs}}$	Sensitivity of TMP prediction

Listed are the evaluations measures used and how they were calculated. Precision and recall for the performance evaluation of the TMH prediction were computed by combining all TMHs within the dataset (i.e., not averaged over each protein). Q_{ok} and Q_{top} were calculated based on all TMPs, where N was the number of TMPs in the dataset, p_i and r_i were the TMH precision and recall for protein i within the dataset, and $t_i = 100\%$ indicated a correctly predicted N-terminal inside/outside topology for protein i.

Dataset New12

Our original datasets had been based on the PDB release from July 2013, when this work began. Shortly before submission of the work in February 2016, that is, 32 months later, we retrieved all TMPs added to OPM and PDBTM since July 2013. We removed all TMPs similar (HVAL > 0) to proteins in datasets used previously (TMP166 and SP1441). Testing the pairwise similarity of the remaining TMPs we found that two pairs were similar (HVAL > 0), but we decided to keep them due to their low HVAL. This resulted in 12 new TMPs (New12 dataset, Supporting Information Table S3) we used for additional testing. Although the statistical power of such a small set is very limited, these 12 constitute the entire addition of completely new structures from 2013/07 to 2016/02. Further, these or structurally related TMPs have most likely not been used to develop any method used for comparison.

Evaluation

As per-protein scores (correct classification as TMP or non-TMP), we compiled the sensitivity (percentage of observed TMPs predicted as TMPs) and the false positive rate (FPR: percentage of soluble proteins predicted as TMPs, Table I). As per-TMH scores (correct identification and placement of TMHs), we compiled the precision (percentage of predicted TMHs that are correct), recall (percentage of observed TMHs predicted as TMHs), Q_{ok} and Q_{top}. Q_{ok} is the percentage of TMPs for which all TMHs are correctly predicted (Table I). Q_{top} requires in addition to Q_{ok} correct topology predictions (in/out: Table I). To resolve conflicts between OPM and PDBTM annotations, we chose whichever fit the prediction best. Note that while sensitivity and recall have the same formula, we used sensitivity in conjunction with TMPs and recall with TMHs to better distinguish between those scores in the text.

Each TMH was considered correctly predicted, if predicted and observed TMH ends were within five residues (Supporting Information Fig. S3), and if predicted and observed TMH overlapped by at least half of the length of the longer of the two helices. These two criteria are more stringent than those that have commonly been used (typically: overlap >3–5 residues anywhere between observed and predicted TMH[36]) and have recently led to re-evaluating TMH prediction methods.[27] None of our major conclusions changed upon applying values slightly different than five residues for the maximum allowed discrepancy between predicted and observed TMH ends (data not shown).

Error rates for the evaluation measures were estimated by bootstrapping,[37] i.e., by re-sampling the population of proteins used for the evaluation 1000 times and calculating the sample standard deviation. Each of these sample populations contained 60% of the original proteins (picked randomly without replacement).

State-of-the-art methods

We compared TMSEG to the best methods,[27] namely to PolyPhobius,[22] MEMSAT3,[25] and MEMSAT-SVM.[26] Like TMSEG, these methods also use evolutionary information to predict TMPs: MEMSAT3 and MEMSAT-SVM automatically generate position-specific scoring matrices (PSSMs) with PSI-BLAST, while PolyPhobius generates multiple sequence alignments (MSAs). To ensure equal conditions for all methods we ran them on our local machines and used the UniProt Reference Cluster with

90% sequence identity (UniRef90, release 2015_03) as the homology search database, i.e., to generate the MSAs or PSSMs. While we used proteins completely unknown to TMSEG to assess its performance, some of the proteins used in our assessment might have been used to develop PolyPhobius, MEMSAT3, or MEMSAT-SVM. In this sense, our assessment was likely to over-estimate their performance, in particular with respect to TMSEG.

Baseline performance

We also compared all methods to a simple baseline predictor similar to TopPred[18]: for all possible segments of 21 consecutive residues, we summed the Eisenberg-hydrophobicity[38] (EisenbergSum, Supporting Information Table S4). All non-overlapping segments with EisenbergSum ≥ 4 were predicted as TMHs, starting with the segments with the highest sum. The inside/outside topology was predicted based on the difference between arginine and lysine residues on either side of the TMHs, i.e., applying Gunnar von Heijne's positive-inside rule.[6,7]

TMSEG input/output

TMSEG needs two input files to successfully run a prediction: a FASTA file with the protein sequence and a PSI-BLAST PSSM file for the input protein. The PSSM file is mandatory and used to include homology-based features that greatly increase the prediction accuracy.

Combining evolutionary information (e.g., PSSMs and MSAs) with machine learning has been the most important improvement in protein prediction and is commonly used in TMH and secondary structure prediction.[24,27,39,40] TMSEG incorporates evolutionary information through PSI-BLAST profiles[41] generated from UniRef90 (release 2015_03). We used two sets of profiles: a training set with a stringent E-value cutoff of 10^{-5} and five iterations for creating the profile, as well as a test set with a less strict E-value cutoff of 10^{-3} and three iterations. We deactivated PSI-BLAST's low-complexity filter and enabled the option to calculate local optimal Smith-Waterman alignments in order to generate longer and more accurate alignments.

In addition, we used biophysical properties (charge, hydrophobicity, polarity; Supporting Information Table S4) and the overall amino acid composition. These features were calculated twice for each residue: once for all substitutions with a positive PSSM score and once based on all substitutions with a negative score.

The standard output gives a brief summary of the positions of the TMHs and signal peptide (if any) and the inside/outside topology. In addition, a raw output is available that also contains the unmodified output probabilities of the machine-learning tools.

TMSEG algorithm

TMSEG combines several machine-learning tools and empirical filters. The machine-learning algorithms used are two random forests (RFs) and one neural network (NN), both of which are implementations from the WEKA Java package.[42] The output of these algorithms is further processed with empirically determined filters and thresholds. The TMSEG algorithm executes four separate steps (Fig. 1):

Step 1: initial per-residue prediction

An RF detects TMHs from the input sequence. This RF slides a window of 19 consecutive residues through the protein sequence, predicting whether or not the central residue in the window is in a TMH, signal peptide, or non-TM region, i.e., the probability of each residue for each state is calculated based on the residue itself and the nine residues left and right of it. For each of the 19 residue positions, we compute the PSSM profile. For the central nine residues in the window, we also compute the average Kyte-Doolittle[43] hydrophobicity, and the percentage of hydrophobic, charged, and polar residues (Supporting Information Table S4).

In addition to these local features, we compile global features: the distance of the residue to the N- and C-terminus, the length of the protein sequence, and the global amino acid composition. The RF assigns three values to each residue corresponding to the probability to be in a TMH, a signal peptide, or a non-TM region. Runtime is decreased by multiplication of the probabilities by 1000 and transformation into integers.

Step 2: per-protein filter: TMP or soluble

The per-residue scores are filtered empirically. First to reduce short peaks of one or two residues, all per-residue scores are smoothed by compiling the median score over five consecutive residues and assigning it to the center residue. Next, each residue is assigned to the state with the highest score (TMH, signal peptide, or non-TM). To prevent over-prediction due to the under-sampling of signal peptide residues, we applied a penalty of 185 (that is, 18.5%) to non-TM and 60 (that is, 6%) to TMH residues. These penalties were optimized during cross-training to best balance over- and under-prediction. Finally, TMHs shorter than seven residues are changed into non-TM regions. If a signal peptide of at least four consecutive residues is identified within the first 40 N-terminal residues ending in residue at position i, TMSEG predicts a signal peptide from residue 1 to residue i ($i \leq 40$). Signal peptide predictions outside the first 40 residues ($i > 40$) are changed into non-TM, but do not invalidate signal peptides inside the first 40 residues.

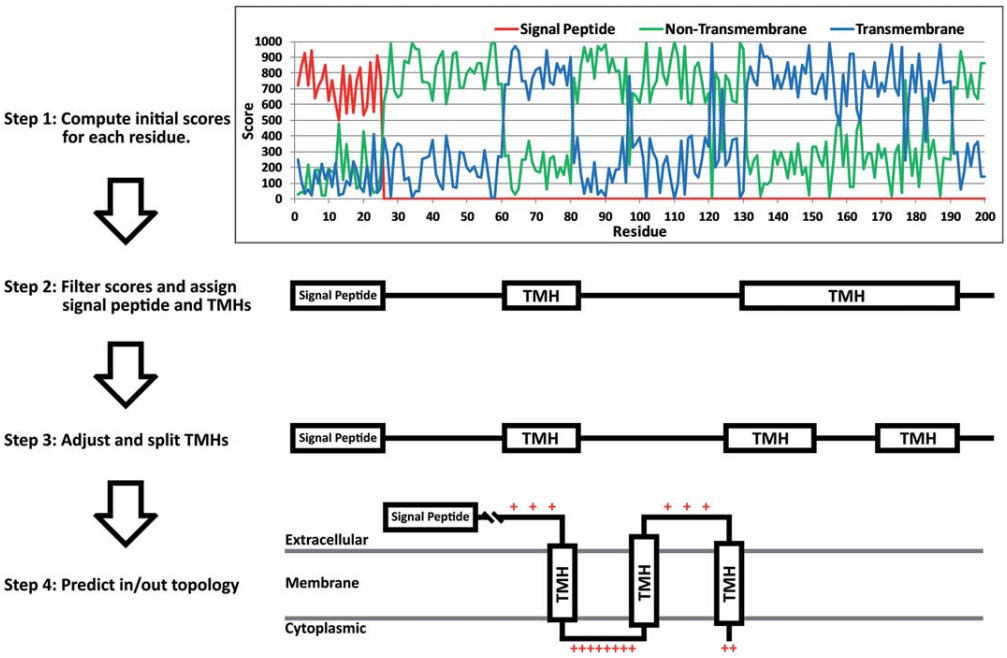


Figure 1

TMSEG algorithm. The new method TMSEG has four steps of machine learning and optimization. Step 1: A random forest (RF) assigns a score to each residue for the three states transmembrane helix (TMH), signal peptide, and non-TM region. Step 2: The previous scores are smoothed (median over 5 residues), all residues are assigned to the state with the highest score, and short segments are removed. Step 3: A segment-based neural network (NN) adjusts the exact position of predicted TMHs, and their length, sometimes splitting TMHs, sometimes shifting, extending, or compressing them. Step 4: The inside/outside topology is predicted by another RF.

Initial predictions with fewer than four consecutive residues are changed into non-TM.

Step 3: refinement of TMHs

In the third step an NN corrects the predicted TMHs. In contrast to the standard sliding window approach of the RF in Step 1, here we introduced a segment-based solution that used as input the following averages over the predicted TMHs: length of predicted TMH, amino acid composition, average hydrophobicity, as well as the percentages of hydrophobic and charged residues. The output of the NN is the predicted probability for the segment to be a TMH. Based on this probability, the predicted TMHs from Step 2 are adjusted.

First, TMHs ≥ 35 residues are split into two TMHs with at least 17 residues, if these two TMHs increase the overall probability. The minimum length of 35 residues for splitting long TMHs and of 17 residues for the resulting two TMHs were empirically chosen based on the overall performance during cross-training. Second, the start and end positions for each TMH are adjusted by shifting them by up to three residues in either direction. Shifts are accepted if they increase the overall probability. The maximum endpoint adjustment by three residues was empirically chosen based on the overall performance during cross-training. In addition, the relatively long minimum TMH lengths to allow splitting and the relatively small shift of maximally three residues of the TMH ends allow TMSEG to maintain a short runtime.

Step 4: topology prediction

Another RF predicts the inside/outside topology of the TMP, i.e., in which direction the TMHs cross the membrane. During this step the non-transmembrane regions are assigned to inside (e.g., cytoplasmic side of the membrane) or outside. This prediction is made for the entire protein. For each TMH, we consider up to 15 residues before and after the TMH, and eight residues at the TMH start and end (for TMHs < 16 these residues overlap). As all predicted TMHs are assumed to cross the membrane, the in/out assignment is switched after each TMH. For each side, we compute as input to the RF the amino acid composition, the percentage of positively charged residues (we consider all arginine and lysine residues), and the absolute difference of positively charged residues between the two sides. Based on the RF output, one side is assigned to be inside (e.g., cytoplasmic), the other to be outside. Residues immediately after predicted signal peptides are assigned to outside (non-cytoplasmic) and all

Table II
Per-Protein Distinction Between Helical TMPs and Other Proteins

Method	TMP sensitivity	TMP FPR	Topology correct	Misclassified in human	More mistakes than TMSEG in human
TMSEG	98 ± 2	3 ± 1	93 ± 4	558	-
PolyPhobius[22]	100 ± 0	5 ± 1	78 ± 7	770	212
MEMSAT3[25]	100 ± 0	28 ± 2	93 ± 4	4313	3755
MEMSAT-SVM[26]	98 ± 2	14 ± 2	88 ± 5	2253	1695
Baseline	95 ± 3	31 ± 2	75 ± 7	5015	4457

Results are provided for all 41 TMPs and 285 soluble proteins in the BlindTest dataset. Error rates are the sample standard deviation based on bootstrapping (cf. Methods). Listed are the *TMP sensitivity* (percentage of correctly predicted helical TMPs), the *TMP FPR* (percentage of non-TMP proteins incorrectly predicted as TMP), *Topology correct* (percentage of proteins for which the topology (inside/outside) was correctly predicted; this differs from Q_{top} which requires topology and all TMHs to be predicted correctly), *Misclassified in human* (estimates the number of proteins misclassfied for the entire human proteome), and *More mistakes than TMSEG in human* (estimates the number of proteins misclassfied more by the method than by TMSEG). The estimates for the human proteome are based on two assumptions: (i) the error estimates on the BlindTest dataset hold true for the human proteome, (ii) the human proteome has 20,196 proteins, 4791 of which are TMPs (cf. Results section "*Application to the human proteome*").

consecutive segments are assigned accordingly without any further prediction.

TMSEG training

To reduce the risk of over-fitting, we split our combined TMP166 and SP1441 datasets into four even splits (cf. Supporting Information Tables S1 and S2). Note that the TMPs from the SP1441 dataset were used to train the random forest in the initial prediction (step 1) as they contain signal peptide annotations. They are, however, not used for the neural network (step 3) or the random forest in step 4, since they have no inside/outside topology annotations and many of their TMH annotations are not supported by experimental evidence.

The first of three splits was used to train, the second to cross-train, i.e., to optimize all other free parameters (e.g., the minimum TMH length), and the last to evaluate performance (test). This procedure was repeated three times, such that each protein had been used exactly once for training, cross-training and testing. The final parameters were frozen according to the overall best performance for all three rotations (on the test set). Given the frozen parameters, we applied the final method to the fourth split, the BlindTest dataset, which had not been used before.

Our careful four-fold split leading to three-fold development (each with training, cross-training, and testing), provided a double protection against overestimating performance. We decided about every detail in the final method before using the BlindTest dataset to evaluate TMSEG as presented here. Many developers use a two-fold split (training/testing), more careful ones the three-fold split (training/cross-training/testing), while the fourth split is occasionally introduced through pre-release data[39] like the New12 dataset that we generated.

RESULTS AND DISCUSSION

The novel TMSEG method introduced here distinguishes between proteins with transmembrane helices (TMHs) and soluble proteins. For all helical transmembrane proteins (TMPs), it predicts the placement of the TMHs, and their orientation in the membrane, i.e., their inside/outside topology. We established sustained performance through cross-validation with two levels of blind testing. We compared our new methods to others, including the best at predicting TMPs,[27] namely PolyPhobius[22] and MEMSAT-SVM.[26] Furthermore, we analyzed MEMSAT3[25] because it excels at the inside/outside topology prediction,[44] and SignalP4.1 as the leading method for signal peptide identification.[35] In addition, we compared to a simple hydrophobicity-based prediction similar to TopPred.[18]

Outstanding per-protein distinction between TMPs and other proteins

TMSEG correctly identified 40 of the 41 TMPs in the BlindTest dataset (98 ± 2% sensitivity) and incorrectly predicted 8 of 285 soluble proteins as TMPs (3 ± 1% false positive rate: FPR). TMSEG performed similar to PolyPhobius (100% sensitivity and 5 ± 1% FPR) and significantly better than MEMSAT3 and MEMSAT-SVM (Table II).

Although signal peptides can be confused with TMHs due to the similarity of their signal, only one of the 8 mistakes of predicting soluble proteins as TMPs originated from incorrectly predicting a signal peptide as a TMH. This shows that training on a dataset containing signal peptides helped significantly to reduce false positive predictions. PolyPhobius, which also includes a sophisticated signal peptide prediction, did not confuse any signal peptides with TMHs. However, MEMSAT-SVM, MEMSAT3, and the Baseline predictor had 13, 41, and 69 predicted TMHs, respectively, that overlapped by at least half their length with annotated signal peptides. Overall, TMSEG was able to reliably detect signal peptides and to not predict them as TMHs (Supporting Information Table S5).

We used the 74 TMPs from the fourth subset of the SP1441 dataset (cf. Supporting Information Table S2) to further test the prediction of signal peptides and TMHs. For these proteins, TMSEG and PolyPhobius incorrectly predicted several single-pass TMPs as soluble proteins, because they confused their TMHs near the N-terminus with signal peptides (Supporting Information Table S5). This trend did not occur with the TMPs from the TMP166 dataset (evident by their high sensitivity values; Table II). An explanation might be that TMPs with TMHs within the first 40 residues are more prevalent in the SP1441 dataset, which makes this misclassification more likely to happen. Although these misclassification rates would lower our previous sensitivity estimates for TMSEG and PolyPhobius (at least for single-pass TMPs with their TMH near the N-terminus), we hesitate to generalize the results to everyday applicability since the SP1441 dataset is biased (it was generated to develop the signal peptide predictor SignalP4.1) and contains many TMPs with a TMH near the N-terminus. Further, only 2 of the 9 TMHs that were incorrectly predicted as SPs had experimental evidence.

While all methods reached high sensitivity, they differed vastly in their false positive rates, i.e., soluble proteins incorrectly considered to contain TMHs (Table II). By translating the error rates, the number of proteins that would be misclassified in the entire human proteome can be estimated using two reasonable assumptions: (i) the error estimates for all methods based on the 326 non-redundant proteins (41 TMPs and 285 soluble proteins) in the BlindTest dataset hold true for the (redundant) human proteome, (ii) the human proteome has 20,196 proteins and 4791 of those are TMPs (cf. Section below "*Application to the human proteome*"). Under these assumptions, TMSEG achieves 97% per-protein accuracy and misclassifies only about 558 human proteins. The second best method, PolyPhobius, makes 770 mistakes (212 more than TMSEG) and MEMSAT-SVM as the third best method already misclassifies 2253 proteins (1695 more than TMSEG, Table II). In fact, TMSEG is almost 8.8-times superior to the Baseline predictor, PolyPhobius over 6.5-times better, and MEMSAT-SVM 2.2-times better than the Baseline predictor (Supporting Information Table S6).

Best overall per-TMH prediction

Overall, TMSEG achieved a sustained level of precision ($87 \pm 3\%$) and recall ($84 \pm 3\%$) for the TMHs, that is, $87 \pm 3\%$ of all predicted TMHs were at the correct position and $84 \pm 3\%$ of all observed TMHs had been accurately predicted [Supporting Information Fig. S4(A,B)]. These values were second to no other method, however, only slightly above the second best method MEMSAT-SVM ($85 \pm 3\%$ precision at $83 \pm 3\%$ recall). All other methods had scores below 80%. For $66 \pm 6\%$ of all TMPs, TMSEG predicted all observed TMHs at their correct positions, i.e., $Q_{ok} = 66 \pm 6\%$ (Fig. 2). MEMSAT-SVM followed as second best with $Q_{ok} = 61 \pm 7\%$ (Fig. 2). Nevertheless, given the small datasets, the top performance of TMSEG remained within one standard deviation of all compared methods, except the baseline hydrophobicity prediction (Fig. 2: error bars).

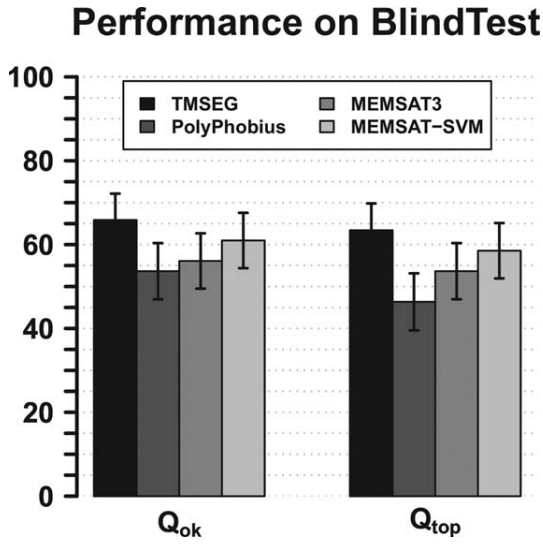


Figure 2

TMSEG compared favorably to state-of-the-art. Results are provided for all 41 TMPs in the BlindTest dataset. Error bars are the sample standard deviation based on bootstrapping (cf. Methods). Shown is on the left the percentage of proteins for which all TMHs were predicted correctly (Q_{ok}, Table I) and on the right the percentage of proteins with correctly predicted TMHs and inside/outside topology (Q_{top}, Table I; note that $Q_{ok} \geq Q_{top}$ by definition).

When comparing the performance on TMP subsets based on the number of TMHs, the performance got worse the more TMHs a protein had [Supporting Information Fig. S4(C,D)]. This might be misunderstood to imply that prediction methods perform better in placing the TMHs in single-pass TMPs than in, e.g., GPCRs (with 7 TMHs). However, this simple numerical comparison ignores the difference in the difficulty of the task: The Baseline predictor reached a high value in Q_{ok} for single-pass TMPs, but failed to predict all TMHs correctly for any TMP with >5 TMHs [Supporting Information Fig. S4(C)]. In fact, when we simply compiled performance for the subset of proteins for which the Baseline predictor failed, we found similar values for proteins with one TMH, those with 2–5, and those with >5 TMHs (Supporting Information Fig. S5).

In contrast, it surprised us that even for the trivial cases, i.e., those for which the Baseline predictor had all TMHs correct, the more advanced methods failed for some of them. This suggests that the large number of different features used by the more advanced methods sometimes interfere with and obscure a strong

hydrophobicity signal. Indeed, only 11 of the 19 trivial TMPs were correctly predicted by all four other methods. However, TMSEG still performed best with $Q_{ok} = 89 \pm 6\%$, followed by MEMSAT3 and MEMSAT-SVM with $Q_{ok} = 84 \pm 7\%$ (data not shown).

Best inside/outside topology prediction

TMSEG and MEMSAT3 correctly placed the N-terminus as inside (e.g., cytoplasmic) or outside (e.g., extracellular), i.e., correctly predicted the topology, for $93 \pm 4\%$ of all TMPs (Table II). When taking into account the global topology and correct TMH placement (i.e., Q_{top}), TMSEG performed better than all other methods reaching $Q_{top} = 63 \pm 6\%$ (Fig. 2). This is five percentage points higher than the second best method, MEMSAT-SVM (albeit still within one standard deviation). Most advanced methods predicted the topology correctly for almost all proteins for which they correctly predicted all TMHs (Q_{top} almost identical to Q_{ok} for all methods, except for the Baseline predictor in Fig. 2).

Application to the human proteome

We applied TMSEG to predict all helical TMPs in the human proteome (20,196 proteins from UniProtKB/Swiss-Prot). TMSEG predicted a total of 5157 TMPs, almost half of these (2300 = 45%) were predicted with one TMH. Given the sensitivity and false positive rate of TMSEG (98 ± 2 and $3 \pm 1\%$, respectively; Table II), we estimate that 462 TMPs were incorrectly predicted (over-predicted) and 96 were missed (under-predicted). In total, we thus misclassified 558 proteins, and our corrected estimate was that humans have about 4791 TMPs, i.e., about 24% of all proteins cross the membrane. While TMSEG misclassified about 558 human proteins, the mistake in the estimate of this percentage appeared to be less than a permille, that is, $\pm 0.01\%$. However, our error estimate might be too simplistic due to the high number of single-pass TMPs for which the error rates are much higher than for proteins with more TMPs.

Confirming previous observations,[2,3] we also observed two peaks of predicted TMPs for proteins with 7 TMHs (819 proteins) and 12 TMHs (189 proteins). These likely represent G protein-coupled receptors (GPCRs) and transporter proteins. Applying UniqueProt to the 5157 predicted TMPs, we found around 500 non-redundant TMPs of which 320 are single-pass TMPs.

Latest experimental structures confirmed our estimates

The 12 new TMPs (New12 dataset) that have recently been added to the PDB constituted the only dataset with truly identical conditions for all methods assessed. The New12 dataset allowed us to confirm the outstanding performance of our new method TMSEG. TMSEG and PolyPhobius correctly identified 10 of the 12 TMPs ($83 \pm 10\%$ sensitivity), while MEMSAT3, MEMSAT-SVM, and the Baseline predictor identified 11 ($92 \pm 7\%$ sensitivity). However, TMSEG correctly predicted every TMH of those 10 TMPs, resulting in a $Q_{ok} = 83 \pm 10\%$, compared to $Q_{ok} = 58 \pm 13\%$ for PolyPhobius, MEMSAT3, and MEMSAT-SVM (Baseline predictor $Q_{ok} = 50 \pm 13\%$). TMSEG also performed best taking into account the topology prediction and reached $Q_{top} = 66 \pm 12\%$, compared to a $Q_{top} = 58 \pm 13\%$ for MEMSAT3 and MEMSAT-SVM, and $Q_{top} = 50 \pm 13\%$ for PolyPhobius and the Baseline predictor.

Comparisons complicated by small datasets

The two small datasets available for evaluation (BlindTest with 41 TMPs and New12 with 12 TMPs) implied high standard errors for many performance estimates. Especially standard errors for the TMH-segment based scores are so high (up to 16 percentage points, Supporting Information Fig. S4) that comparisons between methods hardly provide statistically significant differences on the TMH-segment level. Nevertheless, TMSEG seemed to perform on par with any existing method. Note that the differences in the distinction between helical TMPs and other proteins in the BlindTest dataset were statistically significant even in considering TMSEG as slightly better than the second best PolyPhobius (Table II).

Further, we could not use a single gold standard, because OPM and PDBTM differed in their TMH annotations: comparing the OPM annotations to the PDBTM annotations (that is, "predicting" one with the other) yielded $Q_{ok} = 56 \pm 7\%$. In other words, if we considered one of those experiment-based annotations as the prediction of the other, the average performance would be similar to that of TMSEG and the other methods. When using only OPM or PDBTM annotations to evaluate the prediction performance, TMSEG still performed excellently (Supporting Information Fig. S6). However, this was also the only comparison in which one other method reached a numerically higher value for a dataset than TMSEG, namely MEMSAT-SVM on the PDBTM annotations. Overall, all predictions agreed more with OPM than with PDBTM annotations (Supporting Information Fig. S6).

Performance best with diverse alignments

TMSEG strongly depends on the evolutionary information taken from PSI-BLAST PSSMs. We recommend using a sufficiently large search database (e.g., UniRef90) to generate the PSSMs. Additionally, redundancy reduction might help (e.g., at 90% pairwise sequence identity as in UniRef90).

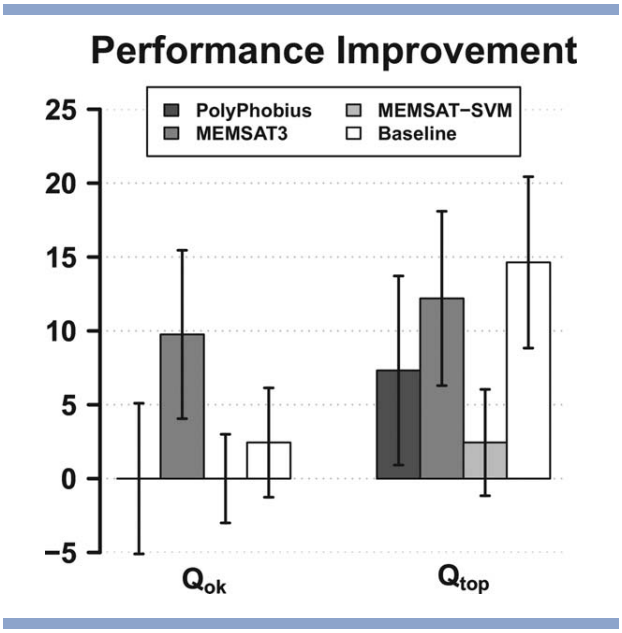


Figure 3

TMSEG applied to refine other methods. The TMSEG algorithm iteratively refines performance through four consecutive steps. Here, we applied Steps 3 and 4 as post-filters to other methods (dataset and error bars as in Fig. 2). Given is the improvement of Q_{ok} and Q_{top} (cf. Table I for definitions) of the prediction method by applying TMSEG, i.e., Q(method + TMSEG) − Q(method). Note that PolyPhobius (first bar on the left) and MEMSAT-SVM (third bar on the left) showed, on average, no improvement in Q_{ok}.

Alignments built from smaller search-databases (e.g., UniRef50 and Swiss-Prot) only slightly lowered the per-protein performance: the sensitivity never dropped below $90 \pm 4\%$, while the false positive rate remained at or below $3 \pm 1\%$. However, the TMH-based precision and recall values dropped substantially (Supporting Information Fig. S7). Thus, for sequences that produce no PSI-BLAST hits, we recommend using a larger search database or—in the rare case that the protein is a true singleton—a method that is independent of evolutionary information, e.g., Phobius.[21,27]

Re-entrant membrane helices not predicted correctly

Our dataset contained only few re-entrant helices, insufficient to learn their prediction (Supporting Information Table S1). Therefore, we considered re-entrant helices as non-TM during training to avoid later interference with the inside/outside topology prediction. Due to the lack of data, we could not reliably assess how well TMSEG distinguishes TMHs from re-entrant membrane helices: The BlindTest dataset included only seven re-entrant regions (OPM and PDBTM annotations combined). TMSEG incorrectly predicted five of seven as TMHs; two of these five were predicted as two separate TMHs; thus, the overall inside/outside topology was not influenced. MEMSAT-SVM, the only tested method that predicts re-entrant helices, identified five of the seven as re-entrant, predicted one as a TMH, and missed the last. When considering re-entrant regions as TMHs, Q_{ok} remained the same for TMSEG and PolyPhobius and dropped by 2–5 percentage points for MEMSAT-SVM, MEMSAT3, and the Baseline predictor.

TMSEG easily combined with other methods

Due to the modularity of TMSEG (i.e., its four separate steps, Fig. 1), it can be used to refine other methods. This includes the adjustment of the TMHs as well as the inside/outside topology prediction. We used the TMH predictions of the reference methods, and applied Steps 3 and 4 of TMSEG to their prediction (Fig. 2). Applying TMSEG as refinement improved the performance for most methods (Fig. 3; Supporting Information Fig. S8). While the improvement was small for the TMH placement (Q_{ok}), TMSEG improved most methods by over eight percentage points in Q_{top} (correct TMHs and topology).

Runtime estimation

We estimated the runtime by applying TMSEG to the human proteome (20,196 proteins). As the time to run PSI-BLAST differs depending on the database size, we decided to use pre-computed PSSMs to measure only the time needed by TMSEG. Given those PSI-BLAST profiles, the prediction for the entire human proteome took about 90 min (Intel Core i7-3632QM 2.2 GHz, 8GB RAM; no multithreading), which corresponds to three to four protein sequences per second.

CONCLUSION

In our hands, our new method TMSEG almost always outperformed existing state-of-the-art prediction methods (Table II, Fig. 2). However, due to the small datasets, many improvements on the per-TMH level remained too small for the large margin of statistical significance (standard errors up to 16 percentage points, Supporting Information Fig. S4). Most importantly, TMSEG achieved the significantly best per-protein classification in the distinction between helical TMPs and all other proteins. For instance, for the prediction of all human proteins, this implied about 558 incorrectly predicted proteins. This number might appear high; however, no method tested reached such a low level, e.g., PolyPhobius misclassified about 200 more proteins than TMSEG and MEMSAT-SVM fared about four times worse (corresponding to >2000 incorrect predictions).

The highest per-protein performance resulted from a combined prediction of TMHs, non-TM regions, and signal peptides. In order to predict re-entrant helices,

another state would have to be introduced; as is, TMSEG predicted five of seven re-entrant helices in our dataset as TMHs. The sustained high levels of per-segment predictions resulted from our new segment-focused algorithm. Another major advantage of our new concept is that it can be used to improve the predictions of most other TMH prediction methods.

Availability and speed

Other than its top performance, using TMSEG may also be recommended due to its speed and because it might help to improve over the method that you run locally. The method is easily and freely available: online through the PredictProtein[45] webserver (www.predictprotein.org), and as standalone Debian package from the Rostlab Debian repository (www.rostlab.org/owiki) and GitHub (www.github.com/Rostlab/TMSEG). A tutorial on how to use PSI-BLAST and TMSEG can be found in the Rostlab Wiki (www.rostlab.org/owiki/index.php/TMSEG).

ACKNOWLEDGMENTS

Thanks to Tim Karl for technical and to Inga Weise (both TUM) for administrative assistance. Thanks to all authors who made their methods openly available and provided us with versions to run on our own machines. Last but not least, thanks to all who practice open science and deposit their data into public databases and those who maintain these excellent databases.

REFERENCES

1. von Heijne G. The membrane protein universe: what's out there and why bother? J Intern Med 2007;261:543–557.
2. Liu J, Rost B. Comparing function and structure between entire proteomes. Protein Sci 2001;10:1970–1979.
3. Fagerberg L, Jonasson K, von Heijne G, Uhlen M, Berglund L. Prediction of the human membrane proteome. Proteomics 2010;10: 1141–1149.
4. Overington JP, Al-Lazikani B, Hopkins AL. How many drug targets are there? Nat Rev Drug Discov 2006;5:993–996.
5. Bigelow HR, Petrey DS, Liu J, Przybylski D, Rost B. Predicting transmembrane beta-barrels in proteomes. Nucleic Acids Res 2004; 32:2566–2577.
6. von Heijne G. The distribution of positively charged residues in bacterial inner membrane proteins correlates with the trans-membrane topology. EMBO J 1986;5:3021–3027.
7. von Heijne G, Gavel Y. Topogenic signals in integral membrane proteins. Eur J Biochem 1988;174:671–678.
8. Punta M, Love J, Handelman S, Hunt JF, Shapiro L, Hendrickson WA, Rost B. Structural genomics target selection for the New York consortium on membrane protein structure. J Struct Funct Genomics 2009;10:255–268.
9. Love J, Mancia F, Shapiro L, Punta M, Rost B, Girvin M, Wang DN, Zhou M, Hunt JF, Szyperski T, Gouaux E, MacKinnon R, McDermott A, Honig B, Inouye M, Montelione G, Hendrickson WA. The New York Consortium on Membrane Protein Structure (NYCOMPS): a high-throughput platform for structural genomics of integral membrane proteins. J Struct Funct Genomics 2010;11: 191–199.
10. Caffrey M. A comprehensive review of the lipid cubic phase or in meso method for crystallizing membrane and soluble proteins and complexes. Acta Crystallogr F Struct Biol Commun 2015;71:3–18.
11. Moraes I, Evans G, Sanchez-Weatherby J, Newstead S, Stewart PD. Membrane protein structure determination - the next generation. Biochim Biophys Acta 2014;1838:78–87.
12. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res 2000;28:235–242.
13. Kloppmann E, Punta M, Rost B. Structural genomics plucks high-hanging membrane proteins. Curr Opin Struct Biol 2012;22:326–332.
14. White SH. Biophysical dissection of membrane proteins. Nature 2009;459:344–346.
15. White SH. The progress of membrane protein structure determination. Protein Sci 2004;13:1948–1949.
16. Lomize MA, Lomize AL, Pogozheva ID, Mosberg HI. OPM: orientations of proteins in membranes database. Bioinformatics 2006;22:623–625.
17. Kozma D, Simon I, Tusnady GE. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. Nucleic Acids Res 2013;41: D524–529.
18. von Heijne G. Membrane protein structure prediction. Hydrophobicity analysis and the positive-inside rule. J Mol Biol 1992;225:487–494.
19. Krogh A, Larsson B, von Heijne G, Sonnhammer EL. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J Mol Biol 2001;305:567–580.
20. Tusnady GE, Simon I. The HMMTOP transmembrane topology prediction server. Bioinformatics 2001;17:849–850.
21. Käll L, Krogh A, Sonnhammer EL. A combined transmembrane topology and signal peptide prediction method. J Mol Biol 2004; 338:1027–1036.
22. Käll L, Krogh A, Sonnhammer EL. An HMM posterior decoder for sequence feature prediction that includes homology information. Bioinformatics 2005;21: i251–257.
23. Rost B, Casadio R, Fariselli P, Sander C. Transmembrane helices predicted at 95% accuracy. Protein Sci 1995;4:521–533.
24. Rost B, Casadio R, Fariselli P. Refining neural network predictions for helical transmembrane proteins by dynamic programming. Proc Int Conf Intell Syst Mol Biol 1996;4:192–200.
25. Jones DT. Improving the accuracy of transmembrane protein topology prediction using evolutionary information. Bioinformatics 2007; 23:538–544.
26. Nugent T, Jones DT. Transmembrane protein topology prediction using support vector machines. BMC Bioinformatics 2009;10:159
27. Reeb J, Kloppmann E, Bernhofer M, Rost B. Evaluation of transmembrane helix predictions in 2014. Proteins 2015;83:473–484.
28. UniProt C. UniProt: a hub for protein information. Nucleic Acids Res 2015;43:D204–212.
29. Velankar S, Dana JM, Jacobsen J, van Ginkel G, Gane PJ, Luo J, Oldfield TJ, O'Donovan C, Martin MJ, Kleywegt GJ. SIFTS: Structure Integration with Function, Taxonomy and Sequences resource. Nucleic Acids Res 2013;41:D483–489.
30. Mika S, Rost B. UniqueProt: Creating representative protein sequence sets. Nucleic Acids Res 2003;31:3789–3791.
31. Sander C, Schneider R. Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins 1991;9:56–68.
32. Rost B. Twilight zone of protein sequence alignments. Protein Eng 1999;12:85–94.
33. Granseth E, Viklund H, Elofsson A. ZPRED: predicting the distance to the membrane center for residues in alpha-helical membrane proteins. Bioinformatics 2006;22:e191–196.
34. Papaloukas C, Granseth E, Viklund H, Elofsson A. Estimating the length of transmembrane helices using Z-coordinate predictions. Protein Sci 2008;17:271–278.
35. Petersen TN, Brunak S, von Heijne G, Nielsen H. SignalP 4.0: discriminating signal peptides from transmembrane regions. Nat Methods 2011;8:785–786.

36. Chen CP, Kernytsky A, Rost B. Transmembrane helix predictions revisited. Protein Sci 2002;11:2774–2791.
37. Efron B, Tibshirani RJ. An introduction to the bootstrap. Chapman & Hall (New York) 1993.
38. Eisenberg D. Three-dimensional structure of membrane and surface proteins. Annu Rev Biochem 1984;53:595–623.
39. Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. Journal of Molecular Biology 1993;232:584–599.
40. Rost B, Liu J, Nair R, Wrzeszczynski KO, Ofran Y. Automatic prediction of protein function. Cellular and Molecular Life Sciences 2003;60:2637–2650.
41. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25:3389–3402.
42. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA Data Mining Software: An Update. SIGKDD Explorations 2009;11:10–18.
43. Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol 1982;157:105–132.
44. Rath EM, Tessier D, Campbell AA, Lee HC, Werner T, Salam NK, Lee LK, Church WB. A benchmark server using high resolution protein structure data, and benchmark results for membrane helix predictions. BMC Bioinformatics 2013;14:111
45. Yachdav G, Kloppmann E, Kajan L, Hecht M, Goldberg T, Hamp T, Honigschmid P, Schafferhans A, Roos M, Bernhofer M, Richter L, Ashkenazy H, Punta M, Schlessinger A, Bromberg Y, Schneider R, Vriend G, Sander C, Ben-Tal N, Rost B. PredictProtein–an open resource for online prediction of protein structural and functional features. Nucleic Acids Res 2014;42:W337–343.

A.1.2. Journal Article: Michael Bernhofer *et al.*, Nucleic Acids Research (2021)

Published online 17 May 2021 *Nucleic Acids Research, 2021, Vol. 49, Web Server issue* ***W535–W540***
https://doi.org/10.1093/nar/gkab354

PredictProtein - Predicting Protein Structure and Function for 29 Years

Michael Bernhofer[1,2,†], Christian Dallago[1,2,*,†], Tim Karl[1,†], Venkata Satagopam[3,4,†], Michael Heinzinger[1,2], Maria Littmann[1,2], Tobias Olenyi[1], Jiajun Qiu[1,5], Konstantin Schütze[1], Guy Yachdav[1], Haim Ashkenazy[6,7], Nir Ben-Tal[8], Yana Bromberg[9], Tatyana Goldberg[1], Laszlo Kajan[10], Sean O'Donoghue[11], Chris Sander[12,13,14], Andrea Schafferhans[1,15], Avner Schlessinger[16], Gerrit Vriend[17], Milot Mirdita[18], Piotr Gawron[3], Wei Gu[3,4], Yohan Jarosz[3,4], Christophe Trefois[3,4], Martin Steinegger[19,20], Reinhard Schneider[3,4] and Burkhard Rost[1,21,22,*]

[1]TUM (Technical University of Munich) Department of Informatics, Bioinformatics & Computational Biology - i12, Boltzmannstr 3, 85748 Garching/Munich, Germany, [2]TUM Graduate School CeDoSIA, Boltzmannstr 11, 85748 Garching, Germany, [3]Luxembourg Centre For Systems Biomedicine (LCSB), University of Luxembourg, Campus Belval, House of Biomedicine II, 6 avenue du Swing, L-4367 Belvaux, Luxembourg, [4]ELIXIR Luxembourg (ELIXIR-LU) Node, University of Luxembourg, Campus Belval, House of Biomedicine II, 6 avenue du Swing, L-4367 Belvaux, Luxembourg, [5]Department of Otolaryngology Head & Neck Surgery, The Ninth People's Hospital & Ear Institute, School of Medicine & Shanghai Key Laboratory of Translational Medicine on Ear and Nose Diseases, Shanghai Jiao Tong University, Shanghai, China, [6]Department of Molecular Biology, Max Planck Institute for Developmental Biology, Tübingen, Germany, [7]The Shmunis School of Biomedicine and Cancer Research, George S. Wise Faculty of Life Sciences, Tel Aviv University, 69978 Tel Aviv, Israel, [8]Department of Biochemistry & Molecular Biology, George S. Wise Faculty of Life Sciences, Tel Aviv University, 69978 Tel Aviv, Israel, [9]Department of Biochemistry and Microbiology, Rutgers University, New Brunswick, NJ 08901, USA, [10]Roche Polska Sp. z o.o., Domaniewska 39B, 02–672 Warsaw, Poland, [11]Garvan Institute of Medical Research, Sydney, Australia, [12]Department of Data Sciences, Dana-Farber Cancer Institute, Boston, MA 02215, USA, [13]Department of Cell Biology, Harvard Medical School, Boston, MA 02215, USA, [14]Broad Institute of MIT and Harvard, Boston, MA 02142, USA, [15]HSWT (Hochschule Weihenstephan Triesdorf | University of Applied Sciences), Department of Bioengineering Sciences, Am Hofgarten 10, 85354 Freising, Germany, [16]Department of Pharmacological Sciences, Icahn School of Medicine at Mount Sinai, New York, NY 10029, USA, [17]BIPS, Poblacion Baco, Mindoro, Philippines, [18]Quantitative and Computational Biology, Max Planck Institute for Biophysical Chemistry, Göttingen, Germany, [19]School of Biological Sciences, Seoul National University, Seoul, South Korea, [20]Artificial Intelligence Institute, Seoul National University, Seoul, South Korea, [21]Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany and [22]TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany

Received February 23, 2021; Revised April 06, 2021; Editorial Decision April 21, 2021; Accepted May 10, 2021

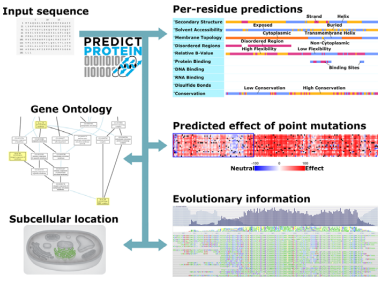
ABSTRACT

Since 1992 *PredictProtein* (https://predictprotein.org) is a one-stop online resource for protein sequence analysis with its main site hosted at the Luxembourg Centre for Systems Biomedicine (LCSB) and queried monthly by over 3,000 users in 2020. *PredictProtein* was the first Internet server for protein predictions. It pioneered combining evolutionary information and machine learning. Given a protein sequence as input, the server outputs multiple sequence alignments, predictions of protein structure in 1D and 2D (secondary structure, solvent accessibility, transmembrane segments, disordered regions, protein flexibility, and disulfide bridges) and predictions of protein function (functional effects of sequence variation or point mutations, Gene Ontology (GO) terms, subcellular localization, and

*To whom correspondence should be addressed. Tel: +49 289 17 811; Email: christian.dallago@tum.de
Correspondence may also be addressed to Burkhard Rost. Email: assistant@rostlab.org
†The authors wish it to be known that, in their opinion, the first four authors should be regarded as joint First Authors.

protein-, RNA-, and DNA binding). *PredictProtein's* infrastructure has moved to the LCSB increasing throughput; the use of MMseqs2 sequence search reduced runtime five-fold (apparently without lowering performance of prediction methods); user interface elements improved usability, and new prediction methods were added. *PredictProtein* recently included predictions from deep learning embeddings (GO and secondary structure) and a method for the prediction of proteins and residues binding DNA, RNA, or other proteins. PredictProtein.org aspires to provide reliable predictions to computational and experimental biologists alike. All scripts and methods are freely available for offline execution in high-throughput settings.

GRAPHICAL ABSTRACT



INTRODUCTION

The sequence is known for far more proteins (1) than experimental annotations of function or structure (2,3). This sequence-annotation gap existed when *PredictProtein* (4,5) started in 1992, and has kept expanding ever since (6). Unannotated sequences contribute crucial evolutionary information to neural networks predicting secondary structure (7,8) that seeded *PredictProtein (PP)* at the European Molecular Biology Laboratory (EMBL) in 1992 (9), the first fully automated, query-driven Internet server providing evolutionary information and structure prediction for any protein. Many other methods predicting aspects of protein function and structure have since joined under the PP roof (4,5,10) now hosted by the Luxembourg Centre of Systems Biomedicine (LCSB).

PP offers an array of structure and function predictions most of which combine machine learning with evolutionary information; now enhanced by a faster alignment algorithm (11,12). A few prediction methods now also use embeddings (13,14) from protein Language Models (LMs) (13–18). Embeddings are much faster to obtain than evolutionary information, yet for many tasks, perform almost as well, or even better (19,20). All PP methods join at PredictProtein.org with interactive visualizations; for some methods, more advanced visualizations are linked (21–23). The reliability of *PredictProtein*, its speed, the continuous integration of well-validated, top methods, and its intuitive interface have attracted thousands of researchers over 29 years of steady operation.

MATERIALS AND METHODS

PredictProtein (PP) provides

multiple sequence alignments (MSAs) and position-specific scoring matrices (PSSMs) computed by a combination of pairwise BLAST (24), PSI-BLAST (25), and MMseqs2 (11,12) on query vs. PDB (26) and query versus UniProt (1). For each residue in the query, the following per-residue predictions are assembled: secondary structure (RePROF/PROFsec (5,27) and ProtBertSec (14)); solvent accessibility (RePROF/PROFacc); transmembrane helices and strands (TMSEG (28) and PROFtmb (29)); protein disorder (Meta-Disorder (30)); backbone flexibility (relative B-values; PROFbval (31)); disulfide bridges (DISULFIND (32)); sequence conservation (ConSurf/ConSeq (33–36)); protein-protein, protein-DNA, and protein-RNA binding residues (ProNA2020 (3)); PROSITE motifs (37); effects of sequence variation (single amino acid variants, SAVs; SNAP2 (38)). For each query per-protein predictions include: transmembrane topology (TMSEG (28)); binary protein-(DNA|RNA|protein) binding (protein binds X or not; ProNA2020 (3)); Gene Ontology (GO) term predictions (goPredSim (19)); subcellular localization (LocTree3 (39)); Pfam (40) domain scans, and some biophysical features. Under the hood, PP computes more results (SOM: PredictProtein Methods; Supplementary Table S1), either as input for frontend methods, or for legacy support.

New: goPredSim embedding-based transfer of Gene Ontology (GO)

goPredSim (19) predicts GO terms by transferring annotations from the most embedding-similar protein. Embeddings are obtained from SeqVec (13); similarity is established through the Euclidean distance between the embedding of a query and a protein with experimental GO annotations. Replicating the conditions of CAFA3 (41) in 2017, goPredSim achieved F_{max} values of 37 ± 2%, 52 ± 2% and 58 ± 2% for BPO (biological process), MFO (molecular function), and CCO (cellular component), respectively (41,42). Using Gene Ontology Annotation (GOA) (43,44) to test 296 proteins annotated after February 2020, goPredSim appeared to reach even slightly higher values that were confirmed by CAFA4 (45).

New: ProtBertSec secondary structure prediction

ProtBertSec predicts secondary structure in three states (helix, strand, other) using ProtBert (14) embeddings derived from training on BFD with almost 3×10^9 proteins (6,46). On a hold-out set from CASP12, ProtBertSec reached a three-state per-residue accuracy of Q3 = 76 ± 1.5% (47). Although below the state-of-the-art (NetSurfP-2.0 (48) at 82%), this method performed on-par with other MSA-based methods, despite itself not using MSAs.

New: ProNA2020 protein–protein, protein–RNA and protein–DNA

ProNA2020 (3) predicts whether or not a protein interacts with other proteins, RNA or DNA (binary), and if so, where

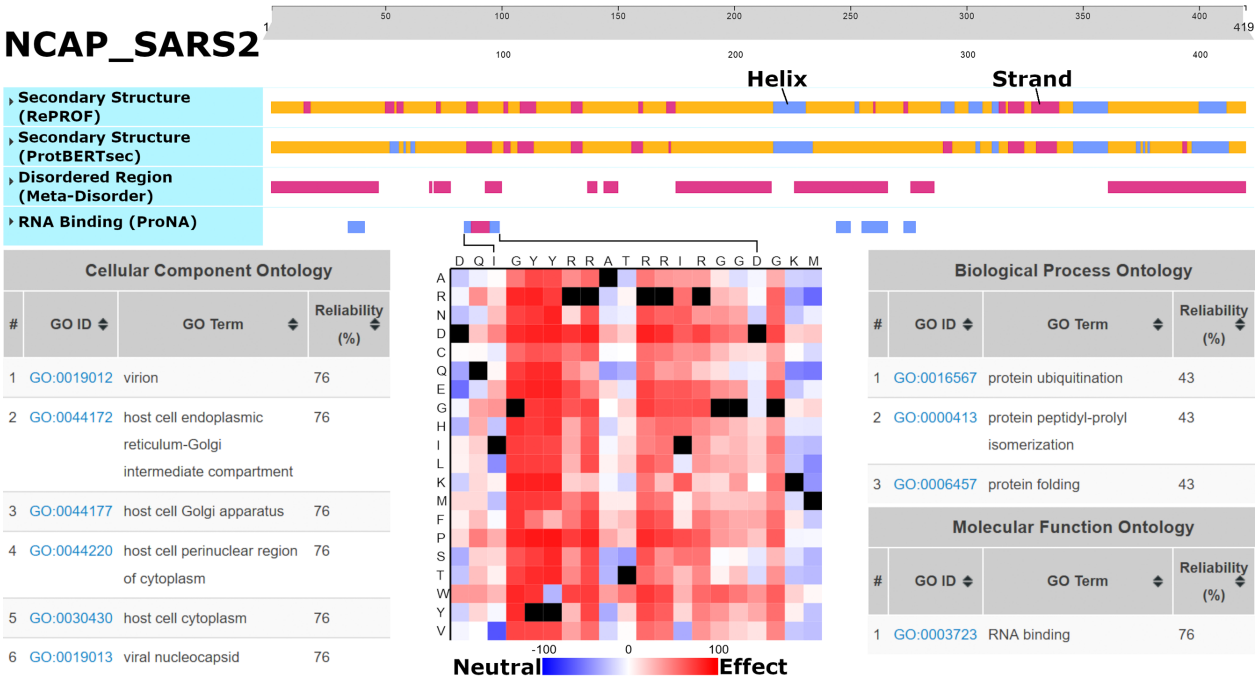


Figure 1. Predictions for SARS-CoV-2 Nucleoprotein (NCAP_SARS2). Underneath the interactive slider at the top: RePROF and ProtBertSec secondary structure (blue: helix; purple: strand; orange: other); Meta-Disorder intrinsically disordered regions (purple); ProNA2020 RNA-binding residues (low confidence: blue; medium confidence: purple). goPredSim transfers of GeneOntology (GO) terms based on embedding similarity (lower left: CCO; lower right: BPO & MFO). SNAP2 predicts the effect of point-mutations on function for the RNA-binding region from I84 to D98 (bottom-center; black: native residue). Link: predictprotein.org/visual_results?req_id=1nAmulUQY$FRPFaP8NTqLW9DzdlTG3B/.

it binds (which residues). The binary per-protein predictions rely on homology and machine learning models employing profile-kernel SVMs (49) and on embeddings from an *in-house* implementation of ProtVec (50). Per-residue predictions (where) use simple neural networks due to data shortage (51–53). ProNA2020 correctly predicted 77 ± 1% of the proteins binding DNA, RNA or protein. In proteins known to bind other proteins, RNA or DNA, ProNA2020 correctly predicted 69 ± 1%, 81 ± 1% and 80 ± 1% of binding residues, respectively.

New: MMseqs2 speedy evolutionary information

Most time-consuming for PP was the search for related proteins in ever growing databases. MMseqs2 (11) finds related sequences blazingly fast and seeds a PSI-BLAST search (25). The query sequence is sent to a dedicated MMseqs2 server that searches for hits against cluster representatives within the UniClust30 (54) and PDB (26) reduced to 70% pairwise percentage sequence identity (PIDE). All hits and their respective cluster members are turned into a MSA and filtered to the 3000 most diverse sequences.

WEB SERVER

Frontend and user interface (UI)

Users query PredictProtein.org by submitting a protein sequence. Results are available in seconds for sequences that had been submitted recently (cf. *PPcache* next section), or within up to 30 min if predictions are recomputed. Per-residue predictions are displayed online via ProtVista (55), which allows to zoom into any sequential protein region (Supplementary Figure S1), and are grouped by category (e.g. secondary structure), which can be expanded to display more detail (e.g. helix, strand, other). On the results page, links to visualize MSAs through *Alignment Viewer* (56) are available. More predictions can be accessed through a menu on the left, e.g. *Gene Ontology Terms*, *Effect of Point Mutations* and *Subcellular Localization*. Prediction views include references and details of outputs, as well as rich visualizations, e.g. GO trees for GO predictions and cell images with highlighted predicted locations for subcellular localization predictions (57).

PPcache, backend and programmatic access

The PP backend lives at LCSB, allowing for up to 48 parallel queries. Results are stored on disc in the *PPcache* (5). Users submitting sequences for which results were over the last two years obtain results immediately. Using the bio-embeddings pipeline (58), the PPcache is enriched by embeddings and embedding-based predictions on the fly. For all methods displayed on the frontend, JSON files compliant with *ProtVista* (55) are available via REST APIs (SOM: Programmatic access), and are in use by external services such as the protein 3D structure visualization suite *Aquaria* (21,23) and by *MolArt* (22).

PredictProtein is available for local use

All results displayed on and downloadable from PP are available through Docker (59) and as source code for local and cloud execution (available at github.com/rostlab).

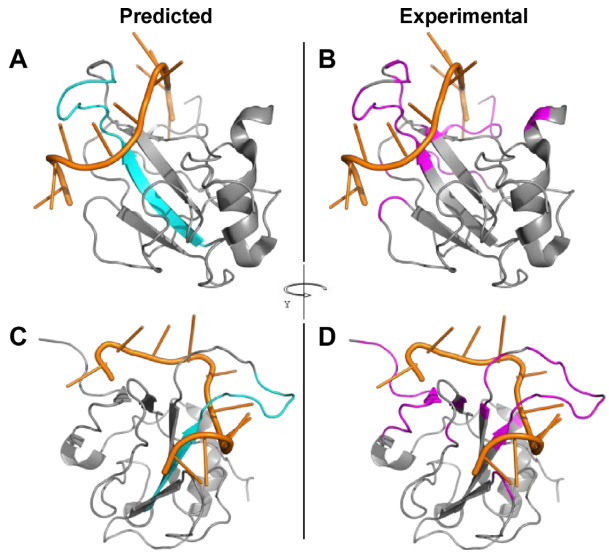


Figure 2. Experimental and predicted RNA-binding residues for NCAP2_SARS2. Predicted (via ProNA2020, in cyan, panels **A** and **C**) and observed (within 5Å, in magenta, panels **B** and **D**) RNA-binding residues for the SARS-CoV-2 nucleoprotein (gray) complexed with a 10-mer ssRNA (orange), PDB structure 7ACT (61). Two-third of the predictions are correct (precision = 0.73, recall = 0.20), which is around the expected average performance reported by ProNA2020. The important sequence consecutive central strand and loop are predicted well, while several short sequence segments that are far away in sequence space but close in structure space are missed, which is expected as ProNA2020 has no notion of 3D structure, i.e., cannot identify 'binding sites'. Panels A and B show a different orientation than panels C and D.

USE CASE

We demonstrate PredictProtein.org tools through predictions of the novel coronavirus SARS-CoV-2 (NCBI:txid2697049) nucleoprotein (UniProt identifier P0DTC9/NCAP_SARS2; Figure 1; SOM: Use Case; Supplementary Figure S2). NCAP_SARS2 has 419 residues and interacts with itself (experimentally verified). Sequence similarity and automatic assignment via UniRule (60) suggest NCAP is RNA-binding (binding with the viral genome), binding with the membrane protein M (UniProt identifier P0DTC5/VME1_SARS2), and is fundamental for virion assembly. goPredSim (19) transferred GO terms from other proteins for MFO (*RNA-binding*; GO:0003723; ECO:0000213) and CCO (compartments in the host cell and viral nucleocapsid; GO:0019013; GO:0044172; GO:0044177; GO:0044220; GO:0030430; ECO:0000255) matching annotations found in UniProt (1). While it missed the experimentally verified MFO term *identical protein binding* (GO:0042802), goPredSim predicted *protein folding* (GO:0006457) and *protein ubiquitination* (GO:0016567) suggesting the nucleoprotein to be involved in biological processes requiring protein binding. ProNA2020 (3) predicts RNA-binding regions, the one with highest confidence between I84 (Isoleucine at position 84) and D98 (Aspartic Acid at 98) (protein sequence in SOM: Use Case). While high resolution experimental data on binding is not available, an NMR structure of the SARS-CoV-2 nucleocapsid phosphoprotein N-terminal domain in complex with 10mer ssRNA (PDB identifier 7ACT (61)) supports the predicted RNA-binding site (Figure 2). Additionally, SNAP2 (38) predicts single amino acid variants (SAVs) in that region to likely affect function, reinforcing the hypothesis that this is a functionally relevant site. Although using different input information (evolutionary vs. embeddings), RePROF (5) and ProtBertSec (14) both predict an unusual content exceeding 70% non-regular (neither helix nor strand) secondary structure, suggesting that most of the nucleoprotein might not form regular structure. This is supported by Meta-Disorder (30) predicting 53% overall disorder. Secondary structure predictions match well high-resolution experimental structures of the nucleoprotein not in complex (e.g., PDB 6VYO (62); 6WJI (63)). Both secondary structure prediction methods managed to zoom into the ordered regions of the protein and predicted e.g., the five beta-strands that are formed within the sequence range I84 (Isoleucine) to A134 (Alanine), and the two helices formed within the sequence range spanned from F346 (Phenylalanine) to T362 (Tyrosine).

CONCLUSION

For almost three decades (preceding Google) *PredictProtein* (PP) has been offering predictions for proteins. PP is the oldest prediction Internet server, online for 6-times as long as most other servers (64–66). It pioneered combining machine learning with evolutionary information and making predictions freely accessible online. While the sequence-annotation gap continues to grow, the sequence-structure gap might be bridged in the near future (67). For the time being, servers such as PP, providing a one-stop solution to predictions from many sustained, novel tools are needed. Now, PP is the first server to offer fast embedding-based predictions of structure (ProtBertSec) and function (goPredSim). By slashing runtime for PSSMs from 72 to 4 min through MMseqs2 and better LCSB hardware, PP also delivers evolutionary information-based predictions fast! Instantaneously if the query is in the precomputed *PPcache*. For heavy use, the offline Docker containers provide predictors out-of-the-box. At no cost to users, *PredictProtein* offers to quickly shine light on proteins for which little is known using well validated prediction methods.

DATA AVAILABILITY

Freely accessible webserver PredictProtein.org; Source and docker images: github.com/rostlab.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

Maintaining *PredictProtein* over three decades has been tough; many colleagues have helped with hands and brains, developers, and users alike. Thanks to all of you! Please find most contributors in Supplementary Table S2 or at predictprotein.org/credits. In particular, thanks to Noua Toukourou and Maharshi Vyas (both LCSB) for invaluable

help with hardware and software; to David Hoksza (Charles U, Prague) for his work on MolArt; to Marco Punta (IRCCS Milano) for his long-term support; to Inga Weise (TUM) for support with many aspects; to Roy Omond (Blue Bubble, Cambridge), Antoine de Daruvar (Univ. Bordeaux), Yanay Ofran (Bar-Ilan Univ.), Jinfeng Liu (Genentech), Tobias Hamp, Maximilian Hecht, Edda Kloppmann (all previously TUM) for contributing methods and code in the past; Johannes Söding for providing resources to develop and maintain MMseqs2.

FUNDING

Michael Bernhofer was supported by the Competence Network for Scientific High Performance Computing in Bavaria [KONWIHR-III BG.DAF]; Christian Dallago is supported by the Deutsche Forschungsgemeinschaft (DFG) [RO 1320/4-1]; Bundesministerium für Bildung und Forschung (BMBF) [031L0168]; Software Campus 2.0 (TU München), BMBF [01IS17049]; Milot Mirdita acknowledges support from the ERC's Horizon 2020 Framework Programme [`Virus-X', project no. 685778]; BMBF CompLifeSci project horizontal4meta. Martin Steinegger acknowledges support from the National Research Foundation of Korea grant funded by the Korean government (MEST) [2019R1A6A1A10073437, NRF-2020M3A9G7103933]; Creative-Pioneering Researchers Program through Seoul National University; Nir Ben-Tal acknowledges the support of Israeli Science Foundation (ISF) [450/16]; Abraham E. Kazan Chair in Structural Biology, Tel Aviv University; Haim Ashkenazy was supported by Humboldt Research Fellowship for Postdoctoral Researchers of the Alexander von Humboldt Foundation; The PredictProtein web server is hosted by ELIXIR-LU, the Luxembourgish node of the European life-science infrastructure. Funding for open access charge: Library of the Technical University of Munich.
Conflict of interest statement. None declared.

REFERENCES

1. The UniProt Consortium (2021) UniProt: the universal protein knowledgebase in 2021. *Nucleic. Acids Res.*, **49**, D480–D489.
2. Rost,B., Liu,J., Nair,R., Wrzeszczynski,K.O. and Ofran,Y. (2003) Automatic prediction of protein function. *Cell. Mol. Life Sci.*, **60**, 2637–2650.
3. Qiu,J., Bernhofer,M., Heinzinger,M., Kemper,S., Norambuena,T., Melo,F. and Rost,B. (2020) ProNA2020 predicts protein–DNA, protein–RNA, and protein–protein binding proteins and residues from sequence. *J. Mol. Biol.*, **432**, 2428–2443.
4. Rost,B. (1996) PHD: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol.*, **266**, 525–539.
5. Yachdav,G., Kloppmann,E., Kajan,L., Hecht,M., Goldberg,T., Hamp,T., Hönigschmid,P., Schafferhans,A., Roos,M., Bernhofer,M. *et al.* (2014) PredictProtein—an open resource for online prediction of protein structural and functional features. *Nucleic Acids Res.*, **42**, W337–W343.
6. Steinegger,M. and Söding,J. (2018) Clustering huge protein sequence sets in linear time. *Nat. Commun.*, **9**, 2542.
7. Rost,B. and Sander,C. (1993) Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Natl. Acad. Sci. U.S.A.*, **90**, 7558–7562.
8. Rost,B. and Sander,C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, **232**, 584–599.
9. Rost,B. and Sander,C. (1992) Jury returns on structure prediction. *Nature*, **360**, 540.
10. Kajan,L., Yachdav,G., Vicedo,E., Steinegger,M., Mirdita,M., Angermuller,C., Bohm,A., Domke,S., Ertl,J., Mertes,C. *et al.* (2013) Cloud prediction of protein structure and function with PredictProtein for Debian. *Biomed. Res. Int.*, **2013**, 398968.
11. Steinegger,M. and Söding,J. (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, **35**, 1026–1028.
12. Mirdita,M., Steinegger,M. and Söding,J. (2019) MMseqs2 desktop and local web server app for fast, interactive sequence searches. *Bioinformatics*, **35**, 2856–2858.
13. Heinzinger,M., Elnaggar,A., Wang,Y., Dallago,C., Nechaev,D., Matthes,F. and Rost,B. (2019) Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, **20**, 723.
14. Elnaggar,A., Heinzinger,M., Dallago,C., Rihawi,G., Wang,Y., Jones,L., Gibbs,T., Feher,T., Angerer,C., Bhowmik,D. *et al.* (2020) ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. arXiv doi: https://arxiv.org/abs/2007.06225, 04 May 2021, preprint: not peer reviewed.
15. Alley,E.C., Khimulya,G., Biswas,S., AlQuraishi,M. and Church,G.M. (2019) Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, **16**, 1315–1322.
16. AlQuraishi,M. (2019) End-to-end differentiable learning of protein structure. *Cell Syst.*, **8**, 292–301.
17. Rao,R., Bhattacharya,N., Thomas,N., Duan,Y., Chen,P., Canny,J., Abbeel,P. and Song,Y. (2019) Evaluating Protein Transfer Learning with TAPE. In: Wallach,H., Larochelle,H., Beygelzimer,A., d'Alché-Buc,F., Fox,E. and Garnett,R. (eds). *Advances in Neural Information Processing Systems*. Vol. **32**. Curran Associates, Inc., pp. 9689–9701.
18. Rives,A., Meier,J., Sercu,T., Goyal,S., Guo,D., Lin,Z., Liu,J., Guo,D., Ott,M., Zitnick,C.L., Ma,J. and Fergus,R. (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA*, **118**, e2016239118.
19. Littmann,M., Heinzinger,M., Dallago,C., Olenyi,T. and Rost,B. (2021) Embeddings from deep learning transfer GO annotations beyond homology. *Sci. Rep.*, **11**, 1160.
20. Rao,R., Ovchinnikov,S., Meier,J., Rives,A. and Sercu,T. (2020) Transformer protein language models are unsupervised structure learners. bioRxiv doi: https://doi.org/10.1101/2020.12.15.422761, 15 December 2020, preprint: not peer reviewed.
21. O'Donoghue,S.I., Sabir,K.S., Kalemanov,M., Stolte,C., Wellmann,B., Ho,V., Roos,M., Perdigão,N., Buske,F.A., Heinrich,J. *et al.* (2015) Aquaria: simplifying discovery and insight from protein structures. *Nat. Methods*, **12**, 98–99.
22. Hoksza,D., Gawron,P., Ostaszewski,M. and Schneider,R. (2018) MolArt: a molecular structure annotation and visualization tool. *Bioinformatics*, **34**, 4127–4128.
23. O'Donoghue,S.I., Schafferhans,A., Sikta,N., Stolte,C., Kaur,S., Ho,B.K., Anderson,S., Procter,J., Dallago,C., Bordin,N. *et al.* (2020) SARS-CoV-2 structural coverage map reveals state changes that disrupt host immunity bioinformatics. bioRxiv doi: https://doi.org/10.1101/2020.07.16.207308, 28 September 2020, preprint: not peer reviewed.
24. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
25. Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
26. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
27. Rost,B. (2001) Protein secondary structure prediction continues to rise. *J. Struct. Biol.*, **134**, 204–218.
28. Bernhofer,M., Kloppmann,E., Reeb,J. and Rost,B. (2016) TMSEG: novel prediction of transmembrane helices. *Proteins*, **84**, 1706–1716.
29. Bigelow,H. and Rost,B. (2006) PROFtmb: a web server for predicting bacterial transmembrane beta barrel proteins. *Nucleic Acids. Res.*, **34**, W186–W188.

30. Schlessinger,A., Punta,M., Yachdav,G., Kajan,L. and Rost,B. (2009) Improved disorder prediction by combination of orthogonal approaches. *PLoS One*, **4**, e4433.
31. Schlessinger,A., Yachdav,G. and Rost,B. (2006) PROFbval: predict flexible and rigid residues in proteins. *Bioinforma. Oxf. Engl.*, **22**, 891–893.
32. Ceroni,A., Passerini,A., Vullo,A. and Frasconi,P. (2006) DISULFIND: a disulfide bonding state and cysteine connectivity prediction server. *Nucleic Acids Res.*, **34**, W177–W181.
33. Berezin,C., Glaser,F., Rosenberg,J., Paz,I., Pupko,T., Fariselli,P., Casadio,R. and Ben-Tal,N. (2004) ConSeq: the identification of functionally and structurally important residues in protein sequences. *Bioinforma. Oxf. Engl.*, **20**, 1322–1324.
34. Ashkenazy,H., Erez,E., Martz,E., Pupko,T. and Ben-Tal,N. (2010) ConSurf 2010: calculating evolutionary conservation in sequence and structure of proteins and nucleic acids. *Nucleic Acids Res.*, **38**, W529–W533.
35. Celniker,G., Nimrod,G., Ashkenazy,H., Glaser,F., Martz,E., Mayrose,I., Pupko,T. and Ben-Tal,N. (2013) ConSurf: using evolutionary data to raise testable hypotheses about protein function. *Isr. J. Chem.*, **53**, 199–206.
36. Ashkenazy,H., Abadi,S., Martz,E., Chay,O., Mayrose,I., Pupko,T. and Ben-Tal,N. (2016) ConSurf 2016: an improved methodology to estimate and visualize evolutionary conservation in macromolecules. *Nucleic Acids Res.*, **44**, W344–W350.
37. Sigrist,C.J.A., de Castro,E., Cerutti,L., Cuche,B.A., Hulo,N., Bridge,A., Bougueleret,L. and Xenarios,I. (2013) New and continuing developments at PROSITE. *Nucleic Acids Res.*, **41**, D344–347.
38. Hecht,M., Bromberg,Y. and Rost,B. (2015) Better prediction of functional effects for sequence variants. *BMC Genomics*, **16 (Suppl 8)**, S1.
39. Goldberg,T., Hecht,M., Hamp,T., Karl,T., Yachdav,G., Ahmed,N., Altermann,U., Angerer,P., Ansorge,S., Balasz,K. *et al.* (2014) LocTree3 prediction of localization. *Nucleic Acids Res.*, **42**, W350–W355.
40. El-Gebali,S., Mistry,J., Bateman,A., Eddy,S.R., Luciani,A., Potter,S.C., Qureshi,M., Richardson,L.J., Salazar,G.A., Smart,A. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.
41. Zhou,N., Jiang,Y., Bergquist,T.R., Lee,A.J., Kacsoh,B.Z., Crocker,A.W., Lewis,K.A., Georghiou,G., Nguyen,H.N., Hamid,M.N. *et al.* (2019) The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol.*, **20**, 244.
42. Jiang,Y., Oron,T.R., Clark,W.T., Bankapur,A.R., D'Andrea,D., Lepore,R., Funk,C.S., Kahanda,I., Verspoor,K.M., Ben-Hur,A. *et al.* (2016) An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.*, **17**, 184.
43. Camon,E., Magrane,M., Barrell,D., Lee,V., Dimmer,E., Maslen,J., Binns,D., Harte,N., Lopez,R. and Apweiler,R. (2004) The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res.*, **32**, D262–D266.
44. Huntley,R.P., Sawford,T., Mutowo-Meullenet,P., Shypitsyna,A., Bonilla,C., Martin,M.J. and O'Donovan,C. (2015) The GOA database: gene ontology annotation updates for 2015. *Nucleic Acids Res.*, **43**, D1057–D1063.
45. El-Mabrouk,N. and Slonim,D.K. (2020) ISMB 2020 proceedings. *Bioinformatics*, **36**, i1–i2.
46. Steinegger,M., Mirdita,M. and Söding,J. (2019) Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods*, **16**, 603–606.
47. Abriata,L.A., Tamò,G.E., Monastyrskyy,B., Kryshtafovych,A. and Peraro,M.D. (2018) Assessment of hard target modeling in CASP12 reveals an emerging role of alignment-based contact prediction methods. *Proteins Struct. Funct. Bioinforma.*, **86**, 97–112.
48. Klausen,M.S., Jespersen,M.C., Nielsen,H., Jensen,K.K., Jurtz,V.I., Sønderby,C.K., Sommer,M.O.A., Winther,O., Nielsen,M., Petersen,B. *et al.* (2019) NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. *Proteins Struct. Funct. Bioinforma.*, **87**, 520–527.
49. Hamp,T., Goldberg,T. and Rost,B. (2013) Accelerating the original profile kernel. *PLoS One*, **8**, e68459.
50. Asgari,E., McHardy,A.C. and Mofrad,M.R.K. (2019) Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). *Sci. Rep.*, **9**, 3577.
51. Norambuena,T. and Melo,F. (2010) The protein-DNA interface database. *BMC Bioinformatics*, **11**, 262.
52. Lewis,B.A., Walia,R.R., Terribilini,M., Ferguson,J., Zheng,C., Honavar,V. and Dobbs,D. (2011) PRIDB: a protein-RNA interface database. *Nucleic. Acids. Res.*, **39**, D277–D282.
53. Hamp,T. and Rost,B. (2015) Evolutionary profiles improve protein-protein interaction prediction from sequence. *Bioinforma. Oxf. Engl.*, **31**, 1945–1950.
54. Mirdita,M., von den Driesch,L., Galiez,C., Martin,M.J., Söding,J. and Steinegger,M. (2017) Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic. Acids. Res.*, **45**, D170–D176.
55. Watkins,X., Garcia,L.J., Pundir,S., Martin,M.J. and Consortium,U. (2017) ProtVista: visualization of protein sequence annotations. *Bioinformatics*, **33**, 2040–2041.
56. Reguant,R., Antipin,Y., Sheridan,R., Dallago,C., Diamantoukos,D., Luna,A., Sander,C. and Gauthier,N.P. (2020) AlignmentViewer: sequence analysis of large protein families. *F1000Research*, **9**, 213.
57. Dallago,C., Goldberg,T., Andrade-Navarro,M.A., Alanis-Lobato,G. and Rost,B. (2020) Visualizing human protein-protein interactions and subcellular localizations on cell images through CellMap. *Curr. Protoc. Bioinforma.*, **69**, e97.
58. Dallago,C., Schütze,K., Heinzinger,M., Olenyi,T., Littmann,M., Lu,A.X., Yang,K.K., Min,S., Yoon,S., Morton,J.T. *et al.* (2021) Learned embeddings from deep learning to visualize and predict protein sets. *Curr. Protoc. Bioinforma.*, **1**, e113.
59. Merkel,D. (2014) Docker: lightweight linux containers for consistent development and deployment. *Linux J.*, **2014**, 2.
60. MacDougall,A., Volynkin,V., Saidi,R., Poggioli,D., Zellner,H., Hatton-Ellis,E., Joshi,V., O'Donovan,C., Orchard,S., Auchincloss,A.H. *et al.* (2020) UniRule: a unified rule resource for automatic annotation in the UniProt Knowledgebase. *Bioinformatics*, **36**, 4643–4648.
61. Dinesh,D.C., Chalupska,D., Silhan,J., Koutna,E., Nencka,R., Veverka,V. and Boura,E. (2020) Structural basis of RNA recognition by the SARS-CoV-2 nucleocapsid phosphoprotein. *PLoS Pathog.*, **16**, e1009100.
62. Chang,C., Michalska,K., Jedrzejczak,R., Maltseva,N., Endres,M., Godzik,A., Kim,Y. and Joachimiak,A. (2020) Crystal structure of RNA binding domain of nucleocapsid phosphoprotein from SARS coronavirus 2. doi:10.2210/pdb6vyo/pdb.
63. Minasov,G., Shuvalova,L., Wiersum,G. and Satchell,K. (2020) 2.05 angstrom resolution crystal structure of C-terminal dimerization domain of nucleocapsid phosphoprotein from SARS-CoV-2. doi:10.2210/pdb6wji/pdb.
64. Schultheiss,S.J., Münch,M.-C., Andreeva,G.D. and Rätsch,G. (2011) Persistence and availability of Web services in computational biology. *PLoS One*, **6**, e24914.
65. Wren,J.D., Georgescu,C., Giles,C.B. and Hennessey,J. (2017) Use it or lose it: citations predict the continued online availability of published bioinformatics resources. *Nucleic Acids. Res.*, **45**, 3627–3633.
66. Kern,F., Fehlmann,T. and Keller,A. (2020) On the lifetime of bioinformatics web services. *Nucleic Acids Res.*, **48**, 12523–12533.
67. Callaway,E. (2020) 'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures. *Nature*, **588**, 203–204.

A.1.3. Journal Article: Michael Bernhofer *et al.*, BMC Bioinformatics (2022)

BMC Bioinformatics

RESEARCH

Open Access

TMbed: transmembrane proteins predicted through language model embeddings

Michael Bernhofer[1,2*] and Burkhard Rost[1,3,4]

*Correspondence:
bernhoferm@rostlab.org

[1] Department of Informatics, Bioinformatics and Computational Biology - i12, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany
[2] TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Boltzmannstr. 11, 85748 Garching, Germany
[3] Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching, Germany
[4] TUM School of Life Sciences Weihenstephan (TUM-WZW), Alte Akademie 8, Freising, Germany

Abstract

Background: Despite the immense importance of transmembrane proteins (TMP) for molecular biology and medicine, experimental 3D structures for TMPs remain about 4–5 times underrepresented compared to non-TMPs. Today's top methods such as AlphaFold2 accurately predict 3D structures for many TMPs, but annotating transmembrane regions remains a limiting step for proteome-wide predictions.

Results: Here, we present TMbed, a novel method inputting embeddings from protein Language Models (pLMs, here ProtT5), to predict for each residue one of four classes: transmembrane helix (TMH), transmembrane strand (TMB), signal peptide, or other. TMbed completes predictions for entire proteomes within hours on a single consumer-grade desktop machine at performance levels similar or better than methods, which are using evolutionary information from multiple sequence alignments (MSAs) of protein families. On the per-protein level, TMbed correctly identified 94 ± 8% of the beta barrel TMPs (53 of 57) and 98 ± 1% of the alpha helical TMPs (557 of 571) in a non-redundant data set, at false positive rates well below 1% (erred on 30 of 5654 non-membrane proteins). On the per-segment level, TMbed correctly placed, on average, 9 of 10 transmembrane segments within five residues of the experimental observation. Our method can handle sequences of up to 4200 residues on standard graphics cards used in desktop PCs (e.g., NVIDIA GeForce RTX 3060).

Conclusions: Based on embeddings from pLMs and two novel filters (Gaussian and Viterbi), TMbed predicts alpha helical and beta barrel TMPs at least as accurately as any other method but at lower false positive rates. Given the few false positives and its outstanding speed, TMbed might be ideal to sieve through millions of 3D structures soon to be predicted, e.g., by AlphaFold2.

Keywords: Protein language models, Protein structure prediction, Transmembrane protein prediction

Background

Structural knowledge of TMPs 4–5 fold underrepresented

Transmembrane proteins (TMP) account for 20–30% of all proteins within any organism [1, 2]; most TMPs cross the membrane with transmembrane helices (TMH). TMPs crossing with transmembrane beta strands (TMB), forming beta barrels, have been estimated to account for 1–2% of all proteins in Gram-negative bacteria; this variety is also

present in mitochondria and chloroplasts [3]. Membrane proteins facilitate many essential processes, including regulation, signaling, and transportation, rendering them targets for most known drugs [4, 5]. Despite this immense relevance for molecular biology and medicine, only about 5% of all three-dimensional (3D) structures in the PDB [6, 7] constitute TMPs [8–10].

Accurate 3D predictions available for proteomes need classification

The prediction of protein structure from sequence leaped in quality through AlphaFold2 [11], Nature's method of the year 2021 [12]. Although AlphaFold2 appears to provide accurate predictions for only very few novel "folds", it importantly increases the width of structural coverage [13]. AlphaFold2 seems to work well on TMPs [14], but for proteome-wide high-throughput studies, we still need to filter out membrane proteins from the structure predictions. Most state-of-the-art (SOTA) TMP prediction methods rely on evolutionary information in the form of multiple sequence alignments (MSA) to achieve their top performance. In our tests we included 13 such methods, namely BetAware-Deep [15], BOCTOPUS2 [16], CCTOP [17, 18], HMM-TM [19–21], OCTOPUS [22], Philius [23], PolyPhobius [24], PRED-TMBB2 [20, 21, 25], PROFtmb [3], SCAMPI2 [26], SPOCTOPUS [27], TMSEG [28], and TOPCONS2 [29].

pLMs capture crucial information without MSAs

Mimicking recent advances of Language Models (LM) in natural language processing (NLP), protein Language Models (pLMs) learn to reconstruct masked parts of protein sequences based on the unmasked local and global information [30–37]. Such pLMs, trained on billions of protein sequences, implicitly extract important information about protein structure and function, essentially capturing aspects of the "language of life" [32]. These aspects can be extracted from the last layers of the deep learning networks into vectors, referred to as embeddings, and used as exclusive input to subsequent methods trained in supervised fashion to successfully predict aspects of protein structure and function [30–34, 36, 38–43]. Often pLM-based methods outperform SOTA methods, which are using evolutionary information on top, and they usually require substantially fewer compute resources. Just before submitting this work, we became aware of another pLM-based TM-prediction method, namely DeepTMHMM [44] using ESM-1b [36] embeddings, and included it in our comparisons.

Here, we combined embeddings generated by the ProtT5 [34] pLM with a simple convolutional neural network (CNN) to create a fast and highly accurate prediction method for alpha helical and beta barrel transmembrane proteins and their overall inside/outside topology. Our new method, TMbed, predicted the presence and location of any TMBs, TMHs, and signal peptides for all proteins of the human proteome within 46 min on our server machine (Additional file 1: Table S1) at the same or better level of performance as other methods, which require substantially more time.

Materials and methods

Data set: membrane proteins (TMPs)

We collected all primary structure files for alpha helical and beta barrel transmembrane proteins (TMP) from OPM [45] and mapped their PDB [6, 7] chain identifiers (PDB-id)

to UniProtKB [46] through SIFTS [47, 48]. Toward this end, we discarded all chimeric chains, all models, and all chains for which OPM failed to map any transmembrane start or end position. This resulted in 2,053 and 206 sequence-unique PDB chains for alpha helical and beta barrel TMPs, respectively.

We used the ATOM coordinates inside the OPM files to assign the inside/outside orientation of sequence segments not within the membrane. We manually inspected inconsistent annotations (e.g., if both ends of a transmembrane segment had the same inside/outside orientation) and cross-referenced them with PDBTM [49–51], PDB, and UniProtKB. We then either corrected such inconsistent annotations or discarded the whole sequence. As OPM does not include signal peptide annotations, we compared our TMP data sets to the set used by SignalP 6.0 [52] and all sequences in UniProtKB/Swiss-Prot with experimentally annotated signal peptides using CD-HIT [53, 54]. For any matches with at least 95% global sequence identity (PIDE), we transferred the signal peptide annotation onto our TMPs. We removed all sequences with fewer than 50 residues to avoid noise from incorrect sequencing fragments, and all sequences with over 15,000 residues to save energy (lower computational costs).

Finally, we removed redundant sequences from the two TMP data sets by clustering them with MMseqs2 [55] to at most 20% local pairwise sequence identity (PIDE) with 40% minimum alignment coverage, i.e., no pair had more than 20% PIDE for any local alignment covering at least 40% of the shorter sequence. The final non-redundant TMP data sets contained 593 alpha helical TMPs and 65 beta barrel TMPs, respectively.

Data set: globular non-membrane proteins

We used the SignalP 6.0 (SP6) dataset for our globular proteins. As the SP6 dataset contained only the first 70 residues of each protein, we took the full sequences from UniProtKB/Swiss-Prot and transferred the signal peptide annotations. To remove any potential membrane proteins from this non-TMP data set, we compared it with CD-HIT [53, 54] against three other data sets: (1) our TMP data sets before redundancy reduction, (2) all protein sequences from UniProtKB/Swiss-Prot with any annotations of transmembrane segments, and (3) all proteins from UniProtKB/Swiss-Prot with any subcellular location annotations for membrane. We removed all proteins from our non-TMP data set with more than 60% global PIDE to any protein in sets 1–3. Again, we dropped all sequences with less than 50 or more than 15,000 residues and applied the same redundancy reduction as before (20% PIDE at 40% alignment coverage). The final non-redundant data set contained 5,859 globular, water-soluble non-TMP proteins; 698 of these have a signal peptide.

Additional redundancy reduction

One anonymous reviewer spotted homologs in our data set after the application of the above protocol. To address this problem, we performed another iteration of redundancy reduction for each of the three data sets using CD-HIT at 20% PIDE. In order to save energy (i.e., avoid retraining our model), we decided to remove clashes for the evaluation, i.e., if two proteins shared more than 20% PIDE, we removed both from the data set (as TMbed was trained on both in the cross-validation protocol). Thereby, this second iteration removed 235 proteins: 8 beta barrel TMPs, 22 alpha helical TMPs, and 205

globular, non-membrane proteins. Our final test data sets included 57 beta barrel TMPs, 571 alpha helical TMPs, and 5654 globular, non-membrane proteins.

Membrane re-entrant regions

Besides transmembrane segments that cross the entire membrane, there are also others, namely membrane segments that briefly enter and exit the membrane on the same side. These are referred to as re-entrant regions [56, 57]. Although rare, some methods explicitly predict them [17, 18, 22, 27, 58]. However, as OPM does not explicitly annotate such regions and since our data set already had a substantial class imbalance between beta barrel TMPs, alpha helical TMPs and, globular proteins, we decided not to predict re-entrant regions.

Embeddings

We generated embeddings with protein Language Models (pLMs) for our data sets using a transformer-based pLM ProtT5-XL-U50 (short: ProtT5) [34]. We discarded the decoder part of ProtT5, keeping only the encoder for increased efficiency (note: encoder embeddings are more informative [34]). The encoder model converts a protein sequence into an embedding matrix that represents each residue in the protein, i.e., each position in the sequence, by a 1024-dimensional vector containing global and local contextualized information. We converted the ProtT5 encoder from 32-bit to 16-bit floating-point format to reduce the memory footprint on the GPU. We took the pre-trained ProtT5 model as is without any further task-specific fine-tuning.

We chose ProtT5 over other embedding models, such as ESM-1b [36], based on our experience with the model and comparisons during previous projects [34, 38]. Furthermore, ProtT5 does not require splitting long sequences, which might remove valuable global context information, while ESM-1b can only handle sequences of up to 1022 residues.

Model architecture

Our TMbed model architecture contained three modules (Additional file 1: Fig. S1): a convolutional neural network (CNN) to generate per-residue predictions, a Gaussian smoothing filter, and a Viterbi decoder to find the best class label for each residue. We implemented the model in PyTorch [59].

Module 1: CNN

The first component of TMbed is a CNN with four layers (Additional file 1: Fig. S1). The first layer is a pointwise convolution, i.e., a convolution with kernel size of 1, which reduces the ProtT5 embeddings for each residue (position in the sequence) from 1024 to 64 dimensions. Next, the model applies layer normalization [60] along the sequence and feature dimensions, followed by a ReLU (Rectified Linear Unit) activation function to introduce non-linearity. The second and third layers consist of two parallel depthwise convolutions; both process the output of the first layer. As depthwise convolutions process each input dimension (feature) independently while considering consecutive residues, those two layers effectively generate sliding weighted sums for each dimension. The kernel sizes of the second and third layer are 9 and 21, respectively, corresponding

to the average length of transmembrane beta strands and helices. As before, the model normalizes the output of both layers and applies the ReLU function. It then concatenates the output of all three layers, constructing a 192-dimensional feature vector for each residue (position in the sequence). The fourth layer is a pointwise convolution combining the outputs from the previous three layers and generates scores for each of the five classes: transmembrane beta strand (B), transmembrane helix (H), signal peptide (S), non-membrane inside (i), and non-membrane outside (o).

Module 2: Gaussian filter

This module smooths the output from the CNN for adjacent residues (sequence positions) to reduce noisy predictions. The filter allows flattening isolated single-residue peaks. For instance, peaks extending of only one to three residues for the classes B and H are often non-informative; similarly short peaks for class S are unlikely correct. The filter uses a Gaussian distribution with standard deviation of 1 and a kernel size of 7, i.e., its seven weights correspond to three standard deviation intervals to the left and right, as well as the central peak. A softmax function then converts the filtered class scores to a class probability distribution.

Module 3: Viterbi decoder

The Viterbi algorithm decodes the class probabilities and assigns a class label to each residue (position in the sequence; Additional file 1: Note S3, Fig. S2). The algorithm uses no trainable parameter; it scores transitions according to the predicted class probabilities. Its purpose is to enforce a simple grammar such that (1) signal peptides can only start at the N-terminus (first residue in protein), (2) signal peptides and transmembrane segments must be at least five residues long (a reasonable trade-off between filtering out false positives and still capturing weak signals), and (3) the prediction for the inside/outside orientation has to change after each transmembrane segment (to simulate crossing through the membrane). Unlike the Gaussian filter, we did not apply the Viterbi decoder during training. This simplified backpropagation and sped up training.

Training details

We performed a stratified five-fold nested cross-validation for model development (Additional file 1: Fig. S3). First, we separated our protein sequences into four groups: beta barrel TMPs, alpha helical TMPs with only a single helix, those with multiple helices, and non-membrane proteins. We further subdivided each group into proteins with and without signal peptides. Next, we randomly and evenly distributed all eight groups into five data sets. As all of our data sets were redundancy reduced, no two splits contained similar protein sequences for any of the classes. However, similarities between proteins of two different classes were allowed, not the least to provide more conservative performance estimates.

During development, we used four of the five splits to create the model and the fifth for testing (Additional file 1: Fig. S3). Of the first four splits, we used three to train the model and the fourth for validation (optimize hyperparameters). We repeated this 3–1 split three more times, each time using a different split for the validation set, and calculated the average performance for every hyperparameter configuration. Next, we trained

a model with the best configuration on all four development splits and estimated its final performance on the independent test split. We performed this whole process a total of five times, each time using a different of the five splits as test data and the remaining four for the development data. This resulted in five final models; each trained, optimized, and tested on independent data sets.

We applied weight decay to all trained weights of the model and added a dropout layer right before the fourth convolutional layer, i.e., the output layer of the CNN. For every training sample (protein sequence), the dropout layer randomly sets 50% of the features to zero across the entire sequence, preventing the model from relying on only a specific subset of features for the prediction.

We trained all models for 15 epochs using the AdamW [61] optimizer and cross-entropy loss. We set the beta parameters to 0.9 and 0.999, used a batch size of 16 sequences, and applied exponential learning rate decay by multiplying the learning rate with a factor of 0.8 every epoch. The initial learning rate and weight decay values were part of the hyperparameters optimized during cross-validation (Additional file 1: Table S2).

The final TMbed model constitutes an ensemble over the five models obtained from the five outer cross-validation iterations (Additional file 1: Fig. S3), i.e., one for each training/test set combination. During runtime, each model generates its own class probabilities (CNN, plus Gaussian filter), which are then averaged and processed by the Viterbi decoder to generate the class labels.

Evaluation and other methods

We evaluated the test performance of TMbed on a per-protein level and on a per-segment level (Additional file 1: Note S1). For protein level statistics, we calculated recall and false positive rate (FPR). We computed those statistics for three protein classes: alpha helical TMPs, beta barrel TMPs, and globular proteins.

We distinguished correct and incorrect segment predictions using two constraints: (1) the observed and predicted segment must overlap such that the intersection of the two is at least half of their union, and (2) neither the start nor the end positions may deviate by more than five residues between the observed and predicted segment (Additional file 1: Fig. S4). All segments predicted meeting both these criteria were considered as "correctly predicted segments", all others as "incorrectly predicted segments". This allowed for a reasonable margin of error regarding the position of a predicted segment, while punishing any gaps introduced into a segment. For per-segment statistics, we calculated recall and precision. We also computed the percentage of proteins with the correct number of predicted segments (Q_{num}), the percentage of proteins for which all segments are correctly predicted (Q_{ok}), and the percentage of correctly predicted segments that also have the correct orientation within the membrane (Q_{top}). We considered only proteins that actually contain the corresponding type of segment when calculating per-segment statistics, e.g., only beta barrel TMPs for transmembrane beta strand segments.

We compared TMbed to other prediction methods for alpha helical and beta barrel TMPs (details in Additional file 1: Note S2): BetAware-Deep [15], BOCTOPUS2 [16], CCTOP [17, 18], DeepTMHMM [44], HMM-TM [19–21], OCTOPUS [22], Philius [23], PolyPhobius [24], PRED-TMBB2 [20, 21, 25], PROFtmb [3], SCAMPI2 [26],

SPOCTOPUS [27], TMSEG [28], and TOPCONS2 [29]. We chose those methods based on their good prediction accuracy and public popularity. For methods predicting only either alpha helical or beta barrel TMPs, we considered the corresponding other type of TMPs as globular proteins for the per-protein statistics. In addition, we generated signal peptide predictions with SignalP 6.0 [52]. The performance of older TMH prediction methods could be triangulated based on previous comprehensive estimate of such methods [28, 62].

Unless stated otherwise, all reported performance values constitute the average performance over the five independent test sets during cross-validation (c.f. *Training details*) and their error margins reflect the 95% confidence interval (CI), i.e., 1.96 times the sample standard error over those five splits (Additional file 1: Tables S5, S6). We considered two values A and B statistically significantly different if they differ by more than their composite 95% confidence interval:

$$|A - B| > CI_c = \sqrt{CI_A^2 + CI_B^2} \tag{1}$$

Additional out-of-distribution benchmark

In the most general sense, machine learning models learn and predict distributions. Most membrane data sets are small and created using the same resources, including OPM [45], PDBTM [49–51], and UniProtKB/Swiss-Prot [46] that often mix experimental annotations with sophisticated algorithms [50, 63–65] to determine the boundaries of transmembrane segments, e.g., by using the 3D structure. Given these constraints, we might expect data sets from different groups to render similar results. Analyzing the validity of this assumption, we included the data set assembled for the development of DeepTMHMM [44]. Three reasons made us chose this set as an alternative perspective: (1) it is recent, (2) it contains helical and beta barrel TMPs, and (3) the authors made their cross-validation predictions available, simplifying comparisons.

We created two distinct data sets from the DeepTMHMM data. First, we collected all proteins common to both data sets (TMbed and DeepTMHMM). We used those proteins to estimate how much the annotations within both data sets agree with each other. In total, there were 1788 proteins common to both data sets: 43 beta barrel TMPs, 184 alpha helical TMPs, 1,560 globular proteins, and one protein (MSPA_MYCS2; Porin MspA) which sits in the outer-membrane of *Mycobacterium smegmatis* [66]. We classified this as beta barrel TMP while DeepTMHMM listed it, most likely incorrectly, as a globular protein. The second data set that we created contained all proteins from the DeepTMHMM data set that were non-redundant to the training data of TMbed. We used PSI-BLAST [67] to find all significant (e-value $< 10^{-4}$) local alignments with a 20% PIDE threshold and 40% alignment coverage to remove the redundant sequences. This second data set contained 667 proteins: 14 beta barrel TMPs, 86 alpha helical TMPs, and 567 globular proteins. We generated predictions with TMbed for those proteins and compared them to the cross-validation predictions for DeepTMHMM, as well as the best performing methods from our own benchmark (CCTOP [17, 18], TOPCONS2 [29], BOCTOPUS2 [16]); we used the DeepTMHMM data set annotations as ground truth.

Data set of new membrane proteins

In order to perform a CASP-like performance evaluation, we gathered all PDB structures published since Feb 05, 2022, which is just after the data for our set and that of DeepTMHMM [44] have been collected. This comprised 1,511 PDB structures (more than 250 of which related to the SARS-CoV-2 protein P0DTD1) that we could map to 1,078 different UniProtKB sequences. We then used PSI-BLAST to remove all sequences similar to our data set or that of DeepTMHMM (e-value $< 10^{-4}$, 20% PIDE at 40% coverage), which resulted in 333 proteins. Next, we predicted transmembrane segments within those proteins using TMbed and DeepTMHMM. For 38 proteins, either TMbed or DeepTMHMM predicted transmembrane segments. After removing any sequences shorter than 100 residues (i.e., fragments) and those in which the predicted segments were not within the resolved regions of the PDB structure, we were left with a set of 5 proteins: one beta barrel TMP and four alpha helical TMPs. Finally, we used the PPM [63–65] algorithm from OPM [45] to estimate the actual membrane boundaries.

Results and discussion

We have developed a new machine learning model, dubbed TMbed; it exclusively uses embeddings from the ProtT5 [34] pLM as input to predict for each residue in a protein sequence to which of the following four "classes" it belongs: transmembrane beta strand (TMB), transmembrane helix (TMH), signal peptide (SP), or non-transmembrane segment. It also predicts the inside/outside orientation of TMBs and TMHs within the membrane, indicating which parts of a protein are inside or outside a cell or compartment. Although the prediction of signal peptides was primarily integrated to improve TMH predictions by preventing the confusion of TMHs with SPs and vice versa, we also evaluated and compared the performance for SP prediction of TMbed to that of other methods.

Reaching SOTA in protein sorting

TMbed detected TMPs with TMHs and TMBs at levels similar or numerically above the best state-of-the-art (SOTA) methods that use evolutionary information from multiple sequence alignments (MSA; Table 1: Recall). Compared to MSA-based methods, TMbed achieved this parity or improvement at a significantly lower false positive rate (FPR), tied only with DeepTMHMM [44], another embedding-based method (Table 1: FPR). Given those numbers, we expect TMbed to misclassify only about 215 proteins for a proteome with 20,000 proteins (Additional file 1: Table S10), e.g., the human proteome, while the other methods would make hundreds more mistakes (DeepTMHMM: 331, TOPCONS2: 683, BOCTOPUS2: 880). Such low FPRs suggest our method as an automated high-throughput filter for TMP detection, e.g., for the creation and annotation of databases, or the decision which AlphaFold2 [11, 68] predictions to parse through advanced software annotating transmembrane regions in 3D structures or predictions [45, 49, 69]. In the binary prediction of whether or not a protein has a signal peptide, TMbed achieved similar levels as the specialist SignalP 6.0 [52] and as DeepTMHMM [44], reaching 99% recall at 0.1% FPR (Additional file 1: Table S3).

Table 1 Per-protein performance. *

	β-TMP (57)		α-TMP (571)		Globular (5654)	
	Recall (%)	**FPR (%)**	**Recall (%)**	**FPR (%)**	**Recall (%)**	**FPR (%)**
TMbed	**93.8 ± 7.5**	***0.1 ± 0.1***	**97.5 ± 0.7**	***0.5 ± 0.2***	***99.5 ± 0.2***	2.8 ± 1.2
DeepTMHMM	77.9 ± 12.7	***0.1 ± 0.1***	95.8 ± 1.3	***0.5 ± 0.2***	***99.5 ± 0.2***	5.9 ± 2.2
TMSEG	–	–	96.5 ± 1.0	2.3 ± 0.3	97.7 ± 0.3	3.5 ± 1.0
TOPCONS2[1]	–	–	94.2 ± 1.3	2.6 ± 0.3	97.4 ± 0.3	5.8 ± 1.3
OCTOPUS[1]	–	–	94.2 ± 1.9	9.1 ± 0.7	90.9 ± 0.7	5.8 ± 1.9
Philius[1]	–	–	92.5 ± 1.4	2.6 ± 0.2	97.4 ± 0.2	7.5 ± 1.4
PolyPhobius[1]	–	–	97.2 ± 1.1	5.3 ± 0.4	94.7 ± 0.4	2.8 ± 1.1
SPOCTOPUS[1]	–	–	**97.5 ± 1.6**	17.2 ± 0.8	82.8 ± 0.8	**2.5 ± 1.6**
SCAMPI2 (MSA)	–	–	94.2 ± 1.6	5.6 ± 0.3	94.4 ± 0.3	5.8 ± 1.6
CCTOP[2]			96.1 ± 2.1	3.7 ± 0.6	96.3 ± 0.6	3.9 ± 2.1
HMM-TM (MSA)[3]	–	–	97.3 ± 1.6	21.4 ± 0.5	78.6 ± 0.5	2.7 ± 1.6
BOCTOPUS2	84.0 ± 13.3	4.2 ± 0.5	–	–	95.8 ± 0.5	16.0 ± 13.3
BetAware-Deep	85.1 ± 9.3	4.7 ± 0.3	–	–	95.3 ± 0.3	14.9 ± 9.3
PRED-TMBB2[4]	88.8 ± 12.1	7.1 ± 0.4	–	–	92.9 ± 0.4	11.2 ± 12.1
PROFtmb	91.9 ± 9.0	6.1 ± 0.5	–	–	93.9 ± 0.5	8.1 ± 9.0

*Evaluation of the ability to distinguish between 57 beta barrel TMPs (β-TMP), 571 alpha helical TMPs (α-TMP) and 5654 globular, water-soluble non-TMP proteins in our data set. Recall and false positive rate (FPR) were averaged over the five independent cross-validation test sets; error margins given for the 95% confidence interval (1.96*standard error); bold: best values for each column; italics: differences statistically significant with over 95% confidence (only computed between best and 2nd best, or all methods ranked 1 and those ranked lower)

[1] Evaluation missing for one of 5,654 globular proteins

[2] Evaluation missing for one of 571 α-TMPs and six of 5,654 globular proteins

[3] Evaluation includes only 51 β-TMPs, 552 α-TMPs, and 5,524 globular proteins due to runtime errors

[4] The local PRED-TMBB2 version did not include the pre-filtering step of the web server. This caused a FPR for β-TMP of almost 78%. Thus, we listed the statistics for the web server predictions, which did not include MSA input

Many of the beta barrel TMPs that prediction methods missed had only two or four transmembrane beta strands (TMB). Such proteins cannot form a pore on their own, instead they have to form complexes with other proteins to function as TMPs, either by binding to other proteins or by forming multimers with additional copies of the same proteins by, e.g., trimerization. In fact, all four beta barrel TMPs missed by TMbed fell into this category. Thus, as all other methods, TMbed performed, on average, worse for beta barrel TMPs that cannot form pores alone. This appeared unsurprising, as the input to all methods were single proteins. For TMPs with TMHs, we also observed lower performance in the distinction between TMP/other for TMPs with a single TMH (recall: 93 ± 3%) compared to those with multiple TMHs (recall: 99 ± 1%). However, TMPs with single helices can function alone.

The embedding-based methods TMbed (introduced here using ProtT5 [34]) and DeepTMHMM [44] (based on ESM-1b [36]) performed at least on par with the SOTA using evolutionary information from MSA (Table 1). While this was already impressive, the real advantage was in the speed. For instance, our method, TMbed, predicted all 6,517 proteins in our data set in about 13 min (i.e., about eight sequences per second) on our server machine (Additional file 1: Table S1); this runtime included generating the ProtT5 embeddings. The other embedding-based method, DeepTMHMM, needed about twice as long (23 min). Meanwhile, methods that search databases and

Table 2 Per-segment performance for TMH (transmembrane helices). *

	TMH (571/2936)				
	Recall (%)	**Precision (%)**	**Q_{ok} (%)**	**Q_{num} (%)**	**Q_{top} (%)**
TMbed	***88.7 ± 0.6***	***88.7 ± 0.7***	***62.4 ± 3.7***	**86.0 ± 2.3**	**96.4 ± 2.7**
DeepTMHMM	80.0 ± 2.4	80.5 ± 2.4	46.2 ± 4.8	85.7 ± 3.5	96.3 ± 2.2
TMSEG	74.5 ± 2.4	77.1 ± 1.7	35.6 ± 2.4	69.9 ± 2.7	83.8 ± 4.7
TOPCONS2	76.4 ± 1.5	78.4 ± 0.8	41.0 ± 3.1	74.4 ± 3.3	91.7 ± 3.1
OCTOPUS	71.6 ± 1.5	75.7 ± 1.4	36.0 ± 2.8	67.6 ± 3.4	87.5 ± 3.1
Philius	70.8 ± 2.2	73.7 ± 0.8	34.2 ± 3.7	66.9 ± 3.4	87.5 ± 2.9
PolyPhobius	76.0 ± 2.1	76.4 ± 1.1	40.3 ± 3.5	74.5 ± 2.8	86.8 ± 2.7
SPOCTOPUS	71.5 ± 1.2	75.8 ± 1.2	35.7 ± 3.3	67.4 ± 5.5	87.2 ± 3.4
SCAMPI2 (MSA)	72.3 ± 2.7	74.1 ± 1.5	33.5 ± 3.0	72.2 ± 4.5	90.6 ± 3.5
CCTOP[1]	77.0 ± 1.7	79.4 ± 1.0	41.9 ± 3.6	82.6 ± 2.7	92.6 ± 2.6
HMM-TM (MSA)[2]	73.3 ± 1.7	72.5 ± 1.2	33.5 ± 1.4	72.1 ± 3.0	88.3 ± 4.2

*Segment performance for transmembrane helix (TMH) prediction based on 571 alpha helical TMPs (α-TMP) with a total of 2936 TMHs. Recall, Precision, Q_{ok}, Q_{num}, and Q_{top} were averaged over the five independent cross-validation test sets; error margins given for the 95% confidence interval (1.96*standard error); bold: best values for each column; italics: differences statistically significant with over 95% confidence (only computed between best and 2nd best).

[1] Evaluation missing for one of 571 α-TMPs.

[2] Evaluation includes only 552 of the 571 α-TMPs due to runtime errors of the method.

Table 3 Per-segment performance for TMB (transmembrane beta strands). *

	TMB (57/768)				
	Recall (%)	**Precision (%)**	**Q_{ok} (%)**	**Q_{num} (%)**	**Q_{top} (%)**
TMbed	**95.0 ± 4.3**	***99.2 ± 0.7***	***80.5 ± 11.4***	**88.1 ± 6.9**	**98.1 ± 3.8**
DeepTMHMM	85.9 ± 6.6	92.5 ± 4.7	46.1 ± 7.6	74.3 ± 13.0	97.2 ± 4.4
BOCTOPUS2	85.3 ± 9.2	96.6 ± 2.0	56.6 ± 18.9	71.2 ± 11.8	98.0 ± 2.0
BetAware-Deep	67.1 ± 6.5	62.2 ± 11.4	8.7 ± 5.3	60.9 ± 14.1	95.7 ± 5.4
PRED-TMBB2 (MSA)	85.4 ± 1.9	75.6 ± 4.8	18.4 ± 15.0	44.5 ± 26.7	95.9 ± 3.4
PROFtmb	78.2 ± 10.1	78.0 ± 6.9	20.2 ± 12.8	46.6 ± 11.7	97.2 ± 1.0

*Segment performance for transmembrane beta strand (TMB) prediction based on 57 beta barrel TMPs (β-TMP) with a total of 768 TMBs. Recall, Precision, Q_{ok}, Q_{num}, and Q_{top} were averaged over the five independent cross-validation test sets; error margins given for the 95% confidence interval (1.96*standard error); bold: best values for each column; italics: differences statistically significant with over 95% confidence (only computed between best and 2nd best)

generate MSAs usually take several seconds or minutes for a single protein sequence [70], or require significant amounts of computing resources (e.g., often more than 100 GB of memory) to achieve comparable runtimes [55].

Excellent transmembrane segment prediction performance

TMbed reached the highest performance for transmembrane segments amongst all methods evaluated (Tables 2, 3). With recall and precision values of 89 ± 1% for TMHs, it significantly outperformed the second best and only other embedding-based method, DeepTMHMM, (80 ± 2%, Table 2). TMbed essentially predicted 62% of all transmembrane helical (TMH) TMPs completely correctly (Q_{ok}, i.e., all TMHs within ± 5 residues of true annotation). DeepTMHMM reached second place with Q_{ok} of 46 ± 4%. This difference between TMbed and DeepTMHMM was over twice that between

DeepTMHMM and the two methods performing third best by this measure, CCTOP [17, 18] and TOPCONS2 [29], which are based on evolutionary information.

The results were largely similar for beta barrel TMPs (TMBs) with TMbed achieving the top performance by all measures: reaching 95% recall and an almost perfect 99% precision. The most pronounced difference was a 23 percentage points lead in Q_{ok} with 80%, compared to BOCTOPUS2 [16] with 57% in second place. Overall, TMbed predicted the correct number of transmembrane segments in 86–88% of TMPs and correctly oriented 98% of TMBs and 96% of TMHs. For signal peptides, TMbed performed on par with SignalP 6.0, reaching 93% recall and 95% precision (Additional file 1: Table S3). For this task, both methods appeared to be slightly outperformed by DeepTMHMM. However, none of those differences exceeded the 95% confidence interval, i.e., the numerically consistent differences were not statistically significant. On top, the signal peptide expert method SignalP 6.0 is the only of the three that distinguishes between different types of signal peptides.

As for the overall per-protein distinction between TMP and non-TMP, the per-segment recall and precision also slightly correlated with the number of transmembrane segments, i.e., the more TMHs or TMBs in a protein the higher the performance (Additional file 1: Table S4). Again, as for the TMP/non-TMP distinction, beta barrel TMPs with only two or four TMBs differed most to those with eight or more.

Gaussian filter and Viterbi decoder improve segment performance

TMbed introduced a Gaussian filter smoothing over some local peaks in the prediction and a Viterbi decoder implicitly enforcing some "grammar-like" rules (Materials & Methods). We investigated the effect of these concepts by comparing the final TMbed architecture to three simpler alternatives: one variant used only the CNN, the other two variants combined the simple CNN with either the Gaussian filter or the Viterbi decoder, not both as TMbed. For the variants without the Gaussian filter, we retrained the CNN using the same hyperparameters but without the filter. Individually, both modules (filter and decoder) significantly improved precision and Q_{ok} for both TMH and TMB, while recall remained largely unaffected (Additional file 1: Table S9). Clearly, either step already improved over just the CNN. However, which of the two was most important depended on the type of TMP: for TMH proteins Viterbi decoder mattered more, for TMB proteins the Gaussian filter. Both steps together performed best throughout without adding any significant overhead to the overall computational costs compared to the other components.

Self-predictions reveal potential membrane proteins

We checked for potential overfitting of our model by predicting the complete data set with the final TMbed ensemble. This meant that four of the five models had seen each of those proteins during training. While the number of misclassified proteins went down, we found that there were still some false predictions, indicating that our models did not simply learn the training data by heart (Additional file 1: Tables S7, S8). In fact, upon closer inspection of the 11 false positive predictions (8 alpha helical and 3 beta barrel TMPs), those appear to be transmembrane proteins incorrectly classified as globular proteins in our data set due to missing annotations in UniProtKB/Swiss-Prot, rather

than incorrect predictions. Two of them, P09489 and P40601, have automatic annotations for an autotransporter domain, which facilitates transport through the membrane. Further, we processed the predicted AlphaFold2 [11, 68] structures of all 11 proteins using the PPM [45] algorithm, which tries to embed 3D structures into a membrane bilayer. For eight of those, the predicted transmembrane segments correlated well with the predicted 3D structures and membrane boundaries (Fig. 1; Additional file 1: Fig. S5). For the other three, the 3D structures and membrane boundaries still indicate transmembrane domains within those proteins, but the predicted transmembrane segments only cover parts of those domains (Additional file 1: Fig. S5, last row). Together, these predictions provided convincing evidence for considering all eleven proteins as TMPs.

Predicting the human proteome in less than an hour

Given that our new method already outperformed the SOTA using evolutionary information from MSAs, the even more important advantage was speed. To estimate prediction throughput, we applied TMbed to all human proteins in 20,375 UniProtKB/Swiss-Prot (version: April 2022; excluding TITIN_HUMAN due to its extreme length of 34,350 residues). Overall, it took our server machine (Additional file 1: Table S1) only 46 min to generate all embeddings and predictions (estimate for consumer-grade PC in the next section). TMbed identified 14 beta barrel TMPs and 4,953 alpha helical TMPs, matching previous estimates for alpha helical TMPs [1, 28]. Two of the 14 TMBs appear to be false positives as TMbed predicted only a single TMB in each protein. The other 12 proteins are either part of the Gasdermin family (A to E), or associated with the mitochondrion, including three proteins for a voltage-dependent anion-selective channel and the TOM40 import receptor.

Further, we generated predictions for all proteins from UniProtKB/Swiss-Prot (version: May 2022), excluding sequences above 10,000 residues (20 proteins). Processing those 566,976 proteins took about 8.5 h on our server machine. TMbed predicted 1,702 beta barrel TMPs and 77,296 alpha helical TMPs (predictions available via our GitHub repository).

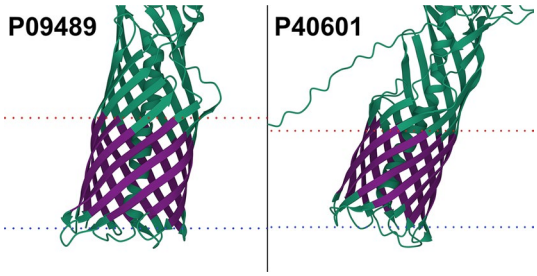


Fig. 1 Potential transmembrane proteins in the globular data set. AlphaFold2 [11, 68] structure of extracellular serine protease (P09489) and Lipase 1 (P40601). Transmembrane segments (dark purple) predicted by TMbed correlate well with membrane boundaries (dotted lines: red = outside, blue = inside) predicted by the PPM [45] web server. Images created using Mol* Viewer [71]. Though our data set lists them as globular proteins, the predicted structures indicate transmembrane domains, which align with segments predicted by our method. The predicted domains overlap with autotransporter domains detected by the UniProtKB [46] automatic annotation system. Transmembrane segment predictions were made with the final TMbed ensemble model

Hardware requirements

Our model needs about 2.5 GB of memory on the GPU when in 16-bit format. The additional memory needed during inference grows with the square of sequence length due to the attention mechanism of the transformer architecture. On our consumer-grade desktop PC (Additional file 1: Table S1), this translated to a maximum sequence length of about 4,200 residues without maxing out the 12 GB of GPU memory. This barred 76 (0.4%) of the 20,376 human proteins from analysis on a personal consumer-hardware solution (NVIDIA GeForce RTX 3060). The prediction (including embedding generation) for 99.6% of the human proteome (20,376 proteins) took about 57 min on our desktop PC. While it is possible to run the model on a CPU, instead of on a GPU, we do not recommend this due to over tenfold larger runtimes. More importantly, the current lack of support of 16-bit floating-point format on CPUs would imply doubling the memory footprint of the model and computations.

Out-of-distribution performance

The two pLM-based methods DeepTMHMM [44] and TMbed appeared to reach similar performance according to the additional out-of-distribution data set (Additional file 1: Tables S11, S12). While DeepTMHMM reached higher scores for beta barrel proteins (Q_{ok} of $79 \pm 22\%$ vs. $64 \pm 26\%$), these were not quite statistically significant. On the other hand, TMbed managed to outperform DeepTMHMM for alpha helical TMPs (Q_{ok} of $53 \pm 11\%$ vs. $47 \pm 10\%$), though again without statistical significance. Furthermore, TMbed performed on par with the OPM baseline (Additional file 1: Table S12), i.e., using the OPM annotations as predictions for the DeepTMHMM data set, implying that TMbed reached its theoretical performance limit on that data set. Surprisingly, TOPCONS2 and CCTOP both outperformed TMbed and DeepTMHMM with Q_{ok} of $65 \pm 10\%$ and $64 \pm 10\%$ (both not statistically significant), respectively.

Taking a closer look at the length distribution for the transmembrane segments in the TMbed and DeepTMHMM data set annotations and predictions (Additional file 1: Fig. S6) revealed differences. First, while the TMB segments in both data sets averaged 9 residues in length, the DeepTMHMM distribution was slightly shifted toward shorter segments (left in Additional file 1: Fig. S6A) but with a wider spread towards longer segments (right in Additional file 1: Fig. S6A). Both of these features were mirrored in the distribution of predicted TMBs. In contrast, the TMH distributions for DeepTMHMM showed an unexpected peak for TMH with 21 residues (both in the annotations used to train DeepTMHMM and in the predictions). In fact, the peak for annotated TMHs at 21 was more than double the value of the two closest length-bins (TMH = 20|22) combined. As the lipid bilayer remains largely invisible in X-ray structures, the exact begin and ends of TMHs may have some errors [28, 45, 49–51, 62]. Thus, when plotting the distribution of TMH length, we expected some kind of normal distribution with a peak around 20-odd residues with more points for longer than for shorter TMHs [72]. In stark contrast to this expectation, the distribution observed for the TMHs used to develop DeepTMHMM appeared to have been obtained through some very different protocol (Additional file 1: Fig. S6B).

In contrast, the distributions for the annotations from OMP and the predictions from TMbed appeared to be more normally distributed with TMH lengths exhibiting a slight

peak at 22 residues. The larger the AI model, the more it succeeds in reproducing features of the development set even when those might be based on less experimentally supported aspects. The DeepTMHMM model reproduced the dubious experimental distribution of TMHs exceedingly (Additional file 1: Fig. S6B, e.g., orange line and bars around peak at 16). Although we do not know the origin of this bias in the DeepTMHMM data set, we have seen similar bias in some prediction methods and automated annotations in UniProtKB/Swiss-Prot. In fact, a quick investigation showed that for 80 of the 184 common alpha helical TMPs the DeepTMHMM annotations matched those found in UniProtKB but not the OPM annotation in our TMbed data set. Of those annotations, 66% (303 of 459) were 21-residues long TMHs, accounting for 73% of all such segments; the other 104 TMPs contained only 19% (114 of 593) TMHs of length 21. This led us to believe that the DeepTMHMM data set contained, in part, length-biased annotations found in UniProtKB. Other examples of methods with length biases include SCAMPI2 and TOPCONS2 that both predicted exclusively TMHs with 21 residues; OCTOPUS and SPOCTOPUS predicted only TMHs of length 15, 21, and 31 (with more than 90% of those being 21 residues). BOCTOPUS2 predicted only beta strands of length 8, 9, and 10, with about 80% of them being nine residues long.

Since TMHs are around 21 residues long, such bias is not necessarily relevant. However, it might point to why performance appears better against some data sets supported less by high-resolution experiments than by others.

Performance on new membrane proteins

Although, the small data set size did not allow for statistically significant results (Additional file 1: Table S13), TMbed performed numerically better than the other methods; in particular, BOCTOPUS2 failed to predict the only beta barrel TMP. While TMbed and DeepTMHMM both missed two of the 30 transmembrane beta strands, TMbed placed the remaining ones, on average, more accurately (recall: 93% vs 87%; precision: 100% vs. 93%). All methods performed worse for the alpha helical TMPs than on the other two benchmark data set, though with a sample size of only four proteins (25 TMHs total), we cannot be sure if this is an effect of testing on novel membrane proteins or simply by chance. Nevertheless, the transmembrane segments predicted by TMbed fit quite well to the membrane boundaries estimated by the PPM [63–65] algorithm (Fig. 2).

No data leakage through pLM

pLMs such as ProtT5 [34] used by TMbed or ESM-1b [36] used by DeepTMHMM are pre-trained on billions of protein sequences. Typically, these include all protein sequences known today. In particular, they include all membrane and non-membrane proteins used in this study. In fact, assuming that the TMPs of known structure account for about 2–5% [78, 79] of all TMPs and that TMPs account for about 20–25% of all proteins, we assume pLMs have been trained on over 490 million TMPs that remain to be experimentally characterized. For the development of AI/ML solutions, it is crucial to establish that methods do not over-fit to existing data but that they will also work for new, unseen data. This implies that in the standard cross-validation process, it is important to not leak any data from development (training and validation used for hyperparameter optimization and model choice)

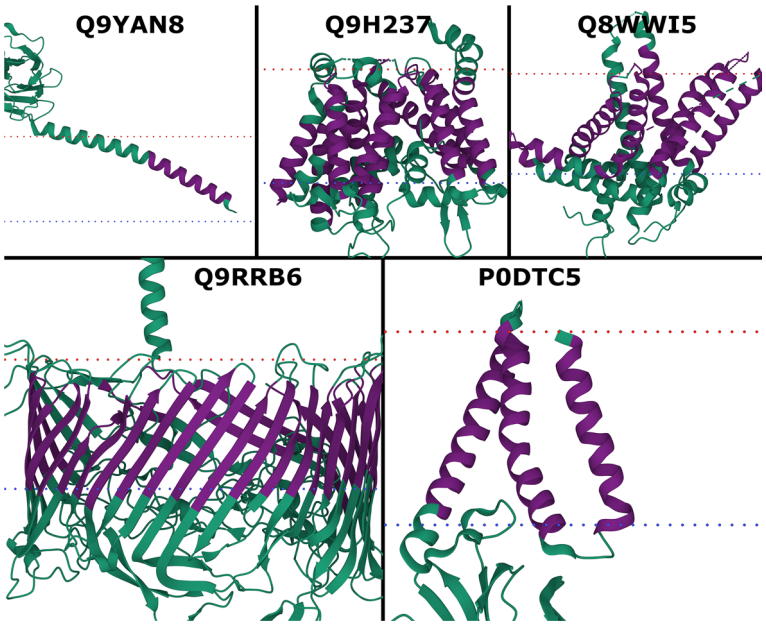


Fig. 2 New membrane proteins. PDB structures for probable flagellin 1 (Q9YAN8; 7TXI [73]), protein-serine O-palmitoleoyltransferase porcupine (Q9H237; 7URD [74]), choline transporter-like protein 1 (Q8WWI5; 7WWB [75]), S-layer protein SlpA (Q9RRB6; 7ZGY [76]), and membrane protein (P0DTC5; 8CTK [77]). Transmembrane segments (dark purple) predicted by TMbed; membrane boundaries (dotted lines: red = outside, blue = inside) predicted by the PPM [45] web server. Images created using Mol* Viewer [71]

to test set (used to assess performance). This implies the necessity for redundancy reduction. This also implies that the conditions for the test set are exactly the same as those that will be encountered in future predictions. For instance, if today's experimental annotations were biased toward bacterial proteins, we might expect performance to be worse for eukaryotic proteins and vice versa.

Both TMbed introduced here and DeepTMHMM are based on the embeddings of pre-trained pLMs; both accomplish the TM-prediction through a subsequent step dubbed transfer learning, in which they use the pLM embeddings as input to train a new AI/ML model in supervised manner on some annotations about membrane segments. Could any data leak from the training of pLMs into the subsequent step of training the TM-prediction methods? Strictly speaking, if no experimental annotations are used, no annotations can leak: the pLMs used here never saw any annotation other than protein sequences.

Even when no annotations could have leaked because none were used for the pLM, should we still ascertain that the conditions for the test set and for the protein for which the method will be applied in the future are identical? We claim that we do not have to ascertain this. However, we cannot support any data for (nor against) this claim. To play devil's advocate, let us assume we had to. The reality is that the vast majority of all predictions likely to be made over the next five years will be for proteins included in these pLMs. In other words, the conditions for future use-cases are exactly the same as those used in our assessment.

Conclusions

TMbed predicts alpha helical (TMH) and beta barrel (TMB) transmembrane proteins (TMPs) with high accuracy (Table 1), performing at least on par or even better than state-of-the-art (SOTA) methods, which depend on evolutionary information from multiple sequence alignments (MSA; Tables 1, 2, 3). In contrast, TMbed exclusively inputs sequence embeddings from the protein language model (pLM) ProtT5. Our novel method shines, in particular, through its low false positive rate (FPR; Table 1), incorrectly predicting fewer than 1% of globular proteins to be TMPs. TMbed also numerically outperformed all other tested methods in terms of correctly predicting transmembrane segments (on average, 9 out of 10 segments were correct; Tables 2, 3). Despite its top performance, the even more significant advantage of TMbed is speed: the high throughput rate of the ProtT5 [34] encoder enables predictions for entire proteomes within an hour, given a suitable GPU (Additional file 1: Table S1). On top, the method runs on consumer-grade GPUs as found in more recent gaming and desktop PCs. Thus, TMbed can be used as a proteome-scale filtering step to scan for transmembrane proteins. Validating the predicted segments with AlphaFold2 [11, 68] structures and the PPM [45] method could be combined into a fast pipeline to discover new membrane proteins, as we have demonstrated with a few proteins. Finally, we provide predictions for 566,976 proteins from UniProtKB/Swiss-Prot (version: May 2022) via our GitHub repository.

Abbreviations

CI	Confidence interval
CNN	Convolutional neural network
MSA	Multiple sequence alignment
OPM	Orientations of proteins in membranes database
PDB	Protein data bank
PDBTM	Protein data bank of transmembrane proteins
pLM	Protein language model
SIFTS	Structure integration with function, taxonomy and sequence
SOTA	State-of-the-art
SP	Signal peptide
TMB	Transmembrane beta strand
TMH	Transmembrane helix
TMP	Transmembrane protein

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-04873-x.

Additional file 1. Supporting Online Material (SOM) containing additional figures, tables and notes.

Acknowledgements

Thanks to Tim Karl and Inga Weise for their help with technical and administrative issues; to Tobias Olenyi, Michael Heinzinger, and Christian Dallago for thoughtful discussions, help with ProtT5, and help with the manuscript; to Konstantinos Tsirigos and Ioannis Tamposis for their support with setting up HMM-TM and PRED-TMBB2; to Pier Luigi Martelli for providing us with BetAware-Deep predictions. Thanks to all who deposit their experimental data in public databases, and to those who maintain them. Last but not least, we thank the reviewers for their constructive criticism, which helped to improve our manuscript.

Author contributions

MB collected the data sets, developed and evaluated the TMbed model, and took the lead in writing the manuscript. BR supervised and guided the work, and co-wrote the manuscript. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. The server machine to run the ProtT5 model was funded by Software Campus Funding (BMBF 01IS17049).

Availability of data and materials

Our code, method, and data sets are freely available in the GitHub repository, https://github.com/BernhoferM/TMbed.

Declarations

Ethical approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 12 June 2022 Accepted: 3 August 2022
Published online: 08 August 2022

References

1. Fagerberg L, Jonasson K, von Heijne G, Uhlen M, Berglund L. Prediction of the human membrane proteome. Proteomics. 2010;10(6):1141–9.
2. Liu J, Rost B. Comparing function and structure between entire proteomes. Protein Sci. 2001;10(10):1970–9.
3. Bigelow HR, Petrey DS, Liu J, Przybylski D, Rost B. Predicting transmembrane beta-barrels in proteomes. Nucleic Acids Res. 2004;32(8):2566–77.
4. Overington JP, Al-Lazikani B, Hopkins AL. How many drug targets are there? Nat Rev Drug Discov. 2006;5(12):993–6.
5. von Heijne G. The membrane protein universe: what's out there and why bother? J Intern Med. 2007;261(6):543–57.
6. ww PDBc. Protein Data Bank: the single global archive for 3D macromolecular structure data. Nucleic Acids Res. 2019;47(D1):D520–D8.
7. Berman H, Henrick K, Nakamura H. Announcing the worldwide Protein Data Bank. Nat Struct Biol. 2003;10(12):980.
8. Hendrickson WA. Atomic-level analysis of membrane-protein structure. Nat Struct Mol Biol. 2016;23(6):464–7.
9. Varga J, Dobson L, Remenyi I, Tusnady GE. TSTMP: target selection for structural genomics of human transmembrane proteins. Nucleic Acids Res. 2017;45(D1):D325–30.
10. Newport TD, Sansom MSP, Stansfeld PJ. The MemProtMD database: a resource for membrane-embedded protein structures and their lipid interactions. Nucleic Acids Res. 2019;47(D1):D390–7.
11. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596(7873):583–9.
12. Marx V. Method of the Year: protein structure prediction. Nat Methods. 2022;19(1):5–10.
13. Bordin N, Sillitoe I, Nallapareddy V, Rauer C, Lam SD, Waman VP, et al. AlphaFold2 reveals commonalities and novelties in protein structure space for 21 model organisms. bioRxiv. 2022:2022.06.02.494367.
14. Hegedus T, Geisler M, Lukacs GL, Farkas B. Ins and outs of AlphaFold2 transmembrane protein structure predictions. Cell Mol Life Sci. 2022;79(1):73.
15. Madeo G, Savojardo C, Martelli PL, Casadio R. BetAware-deep: an accurate web server for discrimination and topology prediction of prokaryotic transmembrane beta-barrel proteins. J Mol Biol. 2021;433(11): 166729.
16. Hayat S, Peters C, Shu N, Tsirigos KD, Elofsson A. Inclusion of dyad-repeat pattern improves topology prediction of transmembrane beta-barrel proteins. Bioinformatics. 2016;32(10):1571–3.
17. Dobson L, Remenyi I, Tusnady GE. The human transmembrane proteome. Biol Direct. 2015;10:31.
18. Dobson L, Remenyi I, Tusnady GE. CCTOP: a Consensus Constrained TOPology prediction web server. Nucleic Acids Res. 2015;43(W1):W408–12.
19. Bagos PG, Liakopoulos TD, Hamodrakas SJ. Algorithms for incorporating prior topological information in HMMs: application to transmembrane proteins. BMC Bioinform. 2006;7:189.
20. Tamposis IA, Sarantopoulou D, Theodoropoulou MC, Stasi EA, Kontou PI, Tsirigos KD, et al. Hidden neural networks for transmembrane protein topology prediction. Comput Struct Biotechnol J. 2021;19:6090–7.
21. Tamposis IA, Theodoropoulou MC, Tsirigos KD, Bagos PG. Extending hidden Markov models to allow conditioning on previous observations. J Bioinform Comput Biol. 2018;16(5):1850019.
22. Viklund H, Elofsson A. OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar. Bioinformatics. 2008;24(15):1662–8.
23. Reynolds SM, Kall L, Riffle ME, Bilmes JA, Noble WS. Transmembrane topology and signal peptide prediction using dynamic bayesian networks. PLoS Comput Biol. 2008;4(11):e1000213.
24. Kall L, Krogh A, Sonnhammer EL. An HMM posterior decoder for sequence feature prediction that includes homology information. Bioinformatics. 2005;21(Suppl 1):i251–7.
25. Tsirigos KD, Elofsson A, Bagos PG. PRED-TMBB2: improved topology prediction and detection of beta-barrel outer membrane proteins. Bioinformatics. 2016;32(17):i665–71.
26. Peters C, Tsirigos KD, Shu N, Elofsson A. Improved topology prediction using the terminal hydrophobic helices rule. Bioinformatics. 2016;32(8):1158–62.
27. Viklund H, Bernsel A, Skwark M, Elofsson A. SPOCTOPUS: a combined predictor of signal peptides and membrane protein topology. Bioinformatics. 2008;24(24):2928–9.
28. Bernhofer M, Kloppmann E, Reeb J, Rost B. TMSEG: Novel prediction of transmembrane helices. Proteins. 2016;84(11):1706–16.
29. Tsirigos KD, Peters C, Shu N, Kall L, Elofsson A. The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. Nucleic Acids Res. 2015;43(W1):W401–7.
30. Asgari E, Mofrad MR. Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS ONE. 2015;10(11):e0141287.

31. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM. Unified rational protein engineering with sequence-based deep representation learning. Nat Methods. 2019;16(12):1315–22.
32. Heinzinger M, Elnaggar A, Wang Y, Dallago C, Nechaev D, Matthes F, et al. Modeling aspects of the language of life through transfer-learning protein sequences. BMC Bioinform. 2019;20(1):723.
33. Bepler T, Berger B. Learning the protein language: evolution, structure, and function. Cell Syst. 2021;12(6):654-69 e3.
34. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, et al. ProtTrans: towards cracking the language of Lifes code through self-supervised deep learning and high performance computing. IEEE Trans Pattern Anal Mach Intell. 2021.
35. Ofer D, Brandes N, Linial M. The language of proteins: NLP, machine learning & protein sequences. Comput Struct Biotechnol J. 2021;19:1750–8.
36. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci USA. 2021;118(15):e2016239118.
37. Wu Z, Johnston KE, Arnold FH, Yang KK. Protein sequence design with deep generative models. Curr Opin Chem Biol. 2021;65:18–27.
38. Marquet C, Heinzinger M, Olenyi T, Dallago C, Erckert K, Bernhofer M, et al. Embeddings from protein language models predict conservation and variant effects. Hum Genet. 2021.
39. Littmann M, Heinzinger M, Dallago C, Weissenow K, Rost B. Protein embeddings and deep learning predict binding residues for various ligand classes. Sci Rep. 2021;11(1):23916.
40. Littmann M, Heinzinger M, Dallago C, Olenyi T, Rost B. Embeddings from deep learning transfer GO annotations beyond homology. Sci Rep. 2021;11(1):1160.
41. Sledzieski S, Singh R, Cowen L, Berger B. D-SCRIPT translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. Cell Syst. 2021;12(10):969-82 e6.
42. Heinzinger M, Littmann M, Sillitoe I, Bordin N, Orengo C, Rost B. Contrastive learning on protein embeddings enlightens midnight zone. bioRxiv. 2022:2021.11.14.468528.
43. Weißenow K, Heinzinger M, Rost B. Protein language model embeddings for fast, accurate, alignment-free protein structure prediction. bioRxiv. 2021:2021.07.31.454572.
44. Hallgren J, Tsirigos KD, Pedersen MD, Almagro Armenteros JJ, Marcatili P, Nielsen H, et al. DeepTMHMM predicts alpha and beta transmembrane proteins using deep neural networks. bioRxiv. 2022:2022.04.08.487609.
45. Lomize MA, Pogozheva ID, Joo H, Mosberg HI, Lomize AL. OPM database and PPM web server: resources for positioning of proteins in membranes. Nucleic Acids Res. 2012;40(Database issue):D370–6.
46. UniProt C. UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Res. 2021;49(D1):D480–9.
47. Dana JM, Gutmanas A, Tyagi N, Qi G, O'Donovan C, Martin M, et al. SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. Nucleic Acids Res. 2019;47(D1):D482–9.
48. Velankar S, Dana JM, Jacobsen J, van Ginkel G, Gane PJ, Luo J, et al. SIFTS: structure integration with function, taxonomy and sequences resource. Nucleic Acids Res. 2013;41(Database issue):D483–9.
49. Kozma D, Simon I, Tusnady GE. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. Nucleic Acids Res. 2013;41(Database issue):D524–9.
50. Tusnady GE, Dosztanyi Z, Simon I. Transmembrane proteins in the Protein Data Bank: identification and classification. Bioinformatics. 2004;20(17):2964–72.
51. Tusnady GE, Dosztanyi Z, Simon I. PDB_TM: selection and membrane localization of transmembrane proteins in the protein data bank. Nucleic Acids Res. 2005;33(Database issue):D275–8.
52. Teufel F, Almagro Armenteros JJ, Johansen AR, Gislason MH, Pihl SI, Tsirigos KD, et al. SignalP 6.0 predicts all five types of signal peptides using protein language models. Nat Biotechnol. 2022.
53. Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics. 2012;28(23):3150–2.
54. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics. 2006;22(13):1658–9.
55. Mirdita M, Steinegger M, Soding J. MMseqs2 desktop and local web server app for fast, interactive sequence searches. Bioinformatics. 2019;35(16):2856–8.
56. Papaloukas C, Granseth E, Viklund H, Elofsson A. Estimating the length of transmembrane helices using Z-coordinate predictions. Protein Sci. 2008;17(2):271–8.
57. Granseth E, Viklund H, Elofsson A. ZPRED: predicting the distance to the membrane center for residues in alpha-helical membrane proteins. Bioinformatics. 2006;22(14):e191–6.
58. Nugent T, Jones DT. Transmembrane protein topology prediction using support vector machines. BMC Bioinform. 2009;10:159.
59. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. 2019.
60. Lei Ba J, Kiros JR, Hinton GE. Layer normalization, 2016 July 01, 2016: arXiv:1607.06450. https://ui.adsabs.harvard.edu/abs/2016arXiv160706450L.
61. Loshchilov I, Hutter F. Decoupled weight decay regularization 2017 November 01, 2017. arXiv:1711.05101. https://ui.adsabs.harvard.edu/abs/2017arXiv171105101L.
62. Reeb J, Kloppmann E, Bernhofer M, Rost B. Evaluation of transmembrane helix predictions in 2014. Proteins. 2015;83(3):473–84.
63. Lomize AL, Pogozheva ID, Mosberg HI. Anisotropic solvent model of the lipid bilayer. 2. Energetics of insertion of small molecules, peptides, and proteins in membranes. J Chem Inf Model. 2011;51(4):930–46.
64. Lomize AL, Pogozheva ID, Lomize MA, Mosberg HI. Positioning of proteins in membranes: a computational approach. Protein Sci. 2006;15(6):1318–33.
65. Lomize AL, Todd SC, Pogozheva ID. Spatial arrangement of proteins in planar and curved membranes by PPM 3.0. Protein Sci. 2022;31(1):209–20.

66. Mahfoud M, Sukumaran S, Hulsmann P, Grieger K, Niederweis M. Topology of the porin MspA in the outer membrane of Mycobacterium smegmatis. J Biol Chem. 2006;281(9):5908–15.
67. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25(17):3389–402.
68. Varadi M, Anyango S, Deshpande M, Nair S, Natassia C, Yordanova G, et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res. 2022;50(D1):D439–44.
69. Tunyasuvunakool K, Adler J, Wu Z, Green T, Zielinski M, Zidek A, et al. Highly accurate protein structure prediction for the human proteome. Nature. 2021;596(7873):590–6.
70. Bernhofer M, Dallago C, Karl T, Satagopam V, Heinzinger M, Littmann M, et al. PredictProtein—predicting protein structure and function for 29 years. Nucleic Acids Res. 2021;49(W1):W535–40.
71. Sehnal D, Bittrich S, Deshpande M, Svobodova R, Berka K, Bazgier V, et al. Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. Nucleic Acids Res. 2021;49(W1):W431–7.
72. Kauko A, Hedin LE, Thebaud E, Cristobal S, Elofsson A, von Heijne G. Repositioning of transmembrane alpha-helices during membrane protein folding. J Mol Biol. 2010;397(1):190–201.
73. Wang F, Cvirkaite-Krupovic V, Baquero DP, Krupovic M, Egelman EH. Cryo-EM of A. pernix flagellum.
74. Liu Y, Qi X, Li X. Catalytic and inhibitory mechanisms of porcupine-mediated Wnt acylation.
75. Xie T, Chi X, Huang B, Ye F, Zhou Q, Huang J. Rational exploration of fold atlas for human solute carrier proteins. Structure. 2022.
76. Farci D, Haniewicz P, de Sanctis D, Iesu L, Kereiche S, Winterhalter M, et al. The cryo-EM structure of the S-layer deinoxanthin-binding complex of Deinococcus radiodurans informs properties of its environmental interactions. J Biol Chem. 2022;298(6):102031.
77. Dolan KA, Kern DM, Kotecha A, Brohawn SG. Cryo-EM structure of SARS-CoV-2 M protein in lipid nanodiscs.
78. Pieper U, Schlessinger A, Kloppmann E, Chang GA, Chou JJ, Dumont ME, et al. Coordinating the impact of structural genomics on the human alpha-helical transmembrane proteome. Nat Struct Mol Biol. 2013;20(2):135–8.
79. Kloppmann E, Punta M, Rost B. Structural genomics plucks high-hanging membrane proteins. Curr Opin Struct Biol. 2012;22(3):326–32.

Publisher's Note