bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

MODELING THE LANGUAGE OF LIFE – DEEP LEARNING PROTEIN SEQUENCES

Michael Heinzinger [1,2,*], Ahmed Elnaggar [1,2,*], Yu Wang [3], Christian Dallago [1,2], Dmitrii Nachaev [1,2], Florian Matthes [4] & Burkhard Rost [1, 5]

1 TUM (Technical University of Munich) Department of Informatics, Bioinformatics & Computational Biology - i12, Boltzmannstr. 3, 85748 Garching/Munich, Germany
2 TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Boltzmannstr. 11, 85748 Garching, Germany
3 Leibniz Supercomputing Centre, Boltzmannstr. 1, 85748 Garching/Munich, Germany
4 TUM Department of Informatics, Software Engineering and Business Information Systems, Boltzmannstr. 1, 85748 Garching/Munich, Germany
5 Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany & TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany & Columbia University, Department of Biochemistry and Molecular Biophysics & New York Consortium on Membrane Protein Structure (NYCOMPS), 701 West, 168th Street, New York, NY 10032, USA
* These authors contributed equally to this work.
Corresponding author: mheinzinger@rostlab.org, http://www.rostlab.org/
Tel: +49-289-17-811 (email rost: assistant@rostlab.org)

Abstract

One common task in Computational Biology is the prediction of aspects of protein function and structure from their amino acid sequence. For 26 years, most state-of-the-art approaches toward this end have been marrying machine learning and evolutionary information resulting from related proteins retrieved at increasing cost from ever growing sequence databases. This search is often so time-consuming to prevent analyzing entire proteomes. On top, evolutionary information is less powerful for smaller families, e.g. for proteins from the *Dark Proteome*. Here, we introduced a novel way to represent protein sequences as continuous vectors (*embeddings*) by utilizing the deep bi-directional language model ELMo that effectively captured the biophysical properties of protein sequences from unlabeled big data (UniRef50). After training, this knowledge was transferred for single protein sequences along with other relevant sequence features. We referred to these new embeddings as *SeqVec* and demonstrated their effectiveness by training comparably simple neural networks on existing data sets for two completely different prediction tasks. For the per-residue level, we predicted secondary structure (for NetSurfP-2.0 data set: Q3=79%±1, Q8=68%±1) and disorder (MCC=0.59±0.03). For the per-protein level, we predicted subcellular localization in ten classes (for DeepLoc dataset: Q10=68%±1) and distinguished membrane-bound from water-soluble proteins (Q2= 87%±1). All results built upon the new tool *SeqVec* derived from single protein sequences. Where the lightning-fast *HHblits* needed on average 0.5 - 5 minutes to generate the evolutionary information for a single protein, our *SeqVec* created the vector representation on average in 0.027 seconds.

Availability: SeqVec: https://github.com/mheinzinger/SeqVec - Predictions: https://embed.protein.properties

Key words: Machine Learning, Language Modelling, Sequence Embedding, Secondary structure prediction, Localization prediction, Transfer Learning, Deep Learning. **Abbreviations used:** **1D**, one-dimensional – information representable in a string such as secondary structure or solvent accessibility; **3D**, three-dimensional; **3D structure**, three-dimensional coordinates of protein structure; **MCC**, Matthews-Correlation-Coefficient; **RSA**, relative solvent accessibility;

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Introduction

Over two decades ago, it was first demonstrated how the combination of evolutionary information, i.e. the profiles extracted from multiple sequence alignments (MSA) of related proteins, and machine learning – then represented by standard feed-forward neural networks (ANN) – completely changed the game of protein secondary structure prediction [1-3]. The concept was quickly followed up by many [4-8] and it was shown how much improvement was possible through using larger families, i.e. by including more diverse evolutionary information [9, 10]. Other objectives that used the same idea included the prediction of transmembrane regions [11-13], solvent accessibility [14], residue flexibility (B-values) [15, 16], inter-residue contacts [17], protein disorder [15, 18-20]. Later, methods predicting aspects of protein function improved through this combination, including predictions of sub-cellular localization (aka cellular location [21, 22]), protein interaction sites [23-25], and the effects of sequence variation upon function [26, 27].

Despite its success, the use of evolutionary information has several limitations: (1) finding and aligning related proteins in large database is computationally expansive. In fact, when databases such as UniProt double every two years [28] even methods as fast as PSI-BLAST [29] have to be replaced by even more efficient solutions such as the “lighting-fast” HHblits [30]. Even the most recent version HHblits3 [31] still needs, on average, several minutes to search UniRef50 (subset of UniProt; ~20% of UniProt at release 2019_02). Even faster solutions such as MMSeqs2 [32] require several hundred GBs of main memory. (2) Evolutionary information is not available for all proteins, e.g. for proteins with substantial intrinsically disordered regions [15, 33, 34], the entire *Dark Proteome* [35], and many less-well studied proteins.

Here, we proposed a novel encoding of protein sequences replacing the explicit search for evolutionary related proteins by an implicit transfer of biophysical information derived from large, unlabeled sequence data (UniRef50). Toward this end, we utilized recent methods that revolutionized Natural Language Processing (NLP), namely the bi-directional language model ELMo [36]. In NLP, these methods are trained on unlabeled text-corpus such as Wikipedia to predict the most probable next word in a sentence, given all previous words in this sentence. By learning a probability distribution for sentences, these models develop autonomously a notion for syntax and semantics of language. The trained vector representations (embeddings) are contextualized, i.e. embeddings of the same word depend on its context. We hypothesized that the ELMo concept could be applied to learn aspects of what makes up the language of life in form of protein sequences. Three main challenges arose. (1) Proteins range from about 30 to 33,000 residues compared to the longest NLP model using 1,024. Longer proteins require more GPU memory and the underlying models (so called LSTMs - Long Short-Term Memory networks [37]) have only a limited capability to remember long-range dependencies. (2) Proteins mostly use 20 standard amino acids and in rare, ambiguous or unknown cases 5 additional characters, compared to up to two million words in NLP. Smaller vocabularies might be problematic if protein sequences encode a similar complexity as sentences. (3) We found UniRef50 to contain almost ten times more tokens (9.5 billion) than the largest existing NLP corpus (1 billion). This increased training time over ten-fold.

We trained the bi-directional language model ELMo on UniRef50. Then we assessed the predictive power of the embeddings by applying them to a variety of prediction tasks on two levels: per-residue (word-level) and per-protein (sentence-level) predictions. Toward per-residue, we implemented the predictions of secondary structure in three (helix, strand, other)

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

and eight states (all DSSP [38]), as well as, of two-state long intrinsic disorder. Toward per-protein, we implemented the predictions of protein subcellular localization in ten classes and a binary classification into membrane-bound and water-soluble proteins. For ease of comparison, we used publicly available datasets from two recent methods that achieved break-through performance through Deep Learning, namely NetSurfP-2.0 (secondary structure [39]) and DeepLoc (localization [39]).

Materials & Methods

Data. UniRef50 training of *SeqVec*: We trained ELMo on UniRef50 [28], a sequence redundancy-reduced subset of the UniProt database clustered at 50% pairwise sequence identity (PIDE). It contained 25 different letters (20 standard and 2 rare amino acids (U and O) plus 3 special cases describing either ambiguous (B, Z) or unknown amino acids (X); Table SOM_1) from 33M proteins with 9,577,889,953 residues. Each protein was treated as a sentence and each amino acid was interpreted as a single word. We referred to the resulting embedding as to *SeqVec*.

Per-residue level: secondary structure & intrinsic disorder (*NetSurfP-2.0*). To simplify compatibility, we used the data set published with a recent method seemingly achieving the top performance of the day in secondary structure prediction, namely *NetSurfP-2.0* [40]. Performance values for the same data set exist also for other recent methods such as *Spider3* [30], *RaptorX* [31, 32] and *JPred4* [33]. The set contains 10,837 sequence-unique (at 25% PIDE) proteins of experimentally know 3D structures from the PDB [34] with a resolution of 2.5 Å (0.25 nm) or better, collected by the PISCES server [35]. DSSP [38] assigned secondary structure and intrinsically disordered residues are flagged (residues without atomic coordinates, i.e. REMARK-465 in the PDB file). The original seven DSSP states (+ 1 for unknown) were mapped upon three states using the common convention: [G,H,I] → H (helix), [B,E] → E (strand), all other to O (other; often misleadingly referred to as *coil* or *loop*). Sequences were extracted from the NetSurfP-2.0 dataset through "Structure Integration with Function, Taxonomy and Sequence" (SIFTS) mapping. Only proteins with identical length in SIFTS and NetSurfP-2.0 were used. This filtering step removed 56 sequences from the training set and three from the test sets (see below: two from CB513, one from CASP12 and none from TS115). We randomly selected 536 (~5%) proteins for early stopping (*cross-training*), leaving 10,256 proteins for training. All published values referred to the following three test sets (also referred to as validation set): **TS115** [37]: 115 proteins from high-quality structures (<3Å) released after 2015 (and at most 30% PIDE to any protein of known structure in the PDB at the time); **CB513** [41]: 513 non-redundant sequences compiled 20 years ago (511 after SIFTS mapping); **CASP12** [39]: 21 proteins taken from the CASP12 free-modelling targets (20 after SIFTS mapping; all 21 fulfilled a stricter criterion toward non-redundancy than the two other sets; non-redundant with respect to all 3D structures known until May 2018 and all their relatives). There are some issues with each of these data sets. Nevertheless, toward our objective of establishing a proof-of-principle, these sets sufficed. All test sets had fewer than 25% PIDE to any protein used for training and cross-training (ascertained by the *NetSurfP-2.0* authors). To compare methods using evolutionary information and those using our new word embeddings, we took the *HHblits* profiles published along with the NetSurfP-2.0 dataset.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Per-protein level: localization & membrane proteins (DeepLoc). Localization prediction was trained and evaluated using the *DeepLoc* dataset [40] for which performance was measured for several methods, namely: LocTree2 [42], MultiLoc2 [43], SherLoc2 [44], CELLO [45], iLoc-Euk [46], WoLF PSORT [47] and YLoc [48]. The dataset contained proteins from UniProtKB/Swiss-Prot [49] (release: 2016_04) with experimental annotation (code: ECO:0000269). The *DeepLoc* authors mapped these to ten classes, removing all proteins with multiple annotations. All these proteins were also classified into *water-soluble* or *membrane-bound* (or as *unknown* if the annotation was ambiguous toward this end). The resulting 13,858 proteins were clustered through PSI-CD-HIT [50, 51] (version 4.0; at 30% PIDE or Eval<10^{-6}). Adding the requirement that the alignment had to cover 80% of the shorter protein, yielded 8,464 clusters. This set was split into training and testing by using the same proteins for testing as the authors of DeepLoc. The training set was randomly sub-divided into 90% for training and 10% for determining early stopping (cross-training set).

Method background ELMo terminology. One-hot encoding (also known as *sparse encoding*) assigns each word (referred to as token in NLP) in the vocabulary an integer N used as the Nth component of a vector with the dimension of the vocabulary size (number of different words). Each component is binary, i.e. either 0 if the word is not present in a sentence/text or 1 if it is. This encoding drove the first application of machine learning that clearly improved over all other methods in protein prediction [1-3]. TF-IDF represents tokens with the product of "frequency of token in data set" times "inverse frequency of token in document". Thereby, rare tokens become more relevant than common words such as "the" (so called *stop words*). This concept resembles that of using k-mers for database searches [29], clustering [52], motifs [53, 54], and prediction methods [42, 47, 55-59]. Context-insensitive word embedding replaced expert features, such as TF-IDF, by algorithms that extracted such knowledge from unlabeled corpus such as Wikipedia, by either predicting the neighboring words, given the center word (skip-gram) or vice versa (CBOW). This became known through *Word2Vec* [60] by Google engineers and showcased for computational biology through *ProtVec* [60, 61]. More specialized implementations are *mut2vec* [18] learning mutations in cancer and *phoscontext2vec* [19] identifying phosphorylation sites. The performance of context-insensitive approaches was pushed to its limits by adding sub-word information (FastText [62]) or global statistics on word co-occurance (GloVe [63]). Context-sensitive word embedding started a new wave of word embedding techniques for NLP in 2018: the particular embedding renders the meaning of *paper tiger* dependent upon the context. Popular examples are ELMo [36] and Bert [22]. In our work, we focused on ELMo that consists of two layers of bidirectional LSTMs and is trained to predict the next word given all previous words in a sentence have achieved state-of-the-art results in several NLP tasks. Both require substantial GPU computing power and time to be trained from scratch, however, ELMo has the advantage that it can handle sequences of variable length. To the best of our knowledge, the context-sensitive ELMo has not been adapted to protein sequences, yet.

Method ELMo adaptation. In order to allow more flexible models and easily integrate into existing solutions, we have used and generated ELMo as word embedding layers. So, no fine-tuning was performed on task-specific sequence sets. Thus, researchers could just replace their current embedding layer with our model to boost their task-specific performance. Furthermore, it will simplify the development of custom models that fit other use-cases. The

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

embedding model takes a protein sequence of arbitrary length and returns 3076 features for each residue in the sequence. These 3076 features were derived by concatenating the outputs of the three internal layers of ELMo, each describing a token with a vector of length 1024. For simplicity, we summed the components of the three 1024-dimensional vectors to form a single 1024-dimensional feature vector describing each residue in a protein. In order to demonstrate the general applicability of SeqVec, we neither fine-tuned the model on a specific prediction task nor optimized the combination of the three internal layers. Instead, we used the standard ELMo configuration [36] with the following changes: (i) reduction to 28 tokens (20 standard and 2 rare (U,O) amino acids + 3 special tokens describing ambiguous (B,Z) or unknown (X) amino acids + 3 special tokens for ELMo indicating the beginning and the end of a sequence), (ii) increase number of unroll steps to 100, (iii) decrease number of negative samples to 20, (iv) increase token number to 9,577,889,953. Our ELMo-like implementation, SeqVec, was trained for three weeks on 5 Nvidia Titan GPUs with 12 GB memory, until it converged with a *perplexity* around 10.5 (Fig. SOM_1).

Method SeqVec 2 prediction. On the per-residue level, the predictive power of the new *SeqVec* embeddings was demonstrated by training a small two-layer Convolutional Neural Network (CNN) in PyTorch using a specific implementation [50] of the ADAM optimizer [51], cross-entropy loss, a learning rate of 0.001 and a batch size of 128 proteins. The first layer (in analogy to the sequence-to-structure network for earlier solutions [1, 2]) consisted of 32-filters each with a sliding window-size of w=7. The second layer (structure-to-structure [1, 2]) created the final predictions by applying again a CNN (w=7) over the output of the first layer. Those two layers were connected through a rectified linear unit (ReLU) and a dropout layer [52] with a dropout-rate of 25% (Fig. 1). This simple architecture was trained independently on three different input combinations each with a different number of free parameters. (i) DeepProf (14,000=14k free parameters): Each residue was described by a vector of size 50 which included a one-hot encoding (20 features), the profiles of evolutionary information (20 features) from HHblits as published previously [40], the state transition probabilities of the Hidden-Markov-Model (7 features) and 3 features describing the local alignment diversity. (ii) DeepSeqVec (232k free parameters): Each protein sequence is represented by the output from SeqVec. The resulting embedding described each residue by a 1024-dimensional vector. (iii) DeepProf+SeqVec (244k free parameters): This model simply concatenated the input vectors used in (i) and (ii). (iv) DeepProtVec (25k free parameters): Each sequence was split into overlapping 3-mers each represented by a 100-dimensional ProtVec [64]. (v) DeepOneHot (7k free parameters): The 20 amino acids were encoded as one-hot vectors as described above. Rare amino acids were mapped to vector with all components set to 0. Consequently, each protein residue was encoded as a 20-dimensional one-hot vector. (vi) DeepBLOSUM65 (8k free parameters): Each protein residue was encoded by its BLOSUM65 substitution matrix [65]. In addition to the 20 standard amino acids, BLOSUM65 also contains substitution scores for the special cases B, Z (ambiguous) and X (unknown), resulting in a feature vector of length 23 for each residue.

On the per-protein level, a simple feed-forward neural network was used to demonstrate the power of the new embeddings. In order to ensure equal-sized input vectors for all proteins, we averaged over the embeddings of all residues in a given protein giving a 1024-dimensional vector for each protein. ProtVec representations were derived on the same way, resulting in a 100-dimensional vector. This vector (either 100-or 1024 dimensional) was

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

compressed to 32 features; the dropout rate was set to 25%, batch normalization [53] and a rectified linear Unit (ReLU) were applied before the final prediction. The resulting network had 33K free parameters (Fig. 4, right panel). We used the following hyper-parameters: learning rate: 0.001, Adam optimizer with cross-entropy loss, batch size: 64. The losses of the individual tasks were summed before backpropagation. Due to the relatively small number of free parameters in our models, the training of all networks completed on a single Nvidia GeForce GTX1080 within a few minutes (11s for the per-protein model, 15m for the DeepSeqVec model).

All models introduced in this work were deliberately designed to be relatively simple to demonstrate the predictive power of SeqVec. More sophisticated architectures using SeqVec are likely to outperform the approaches introduced here.

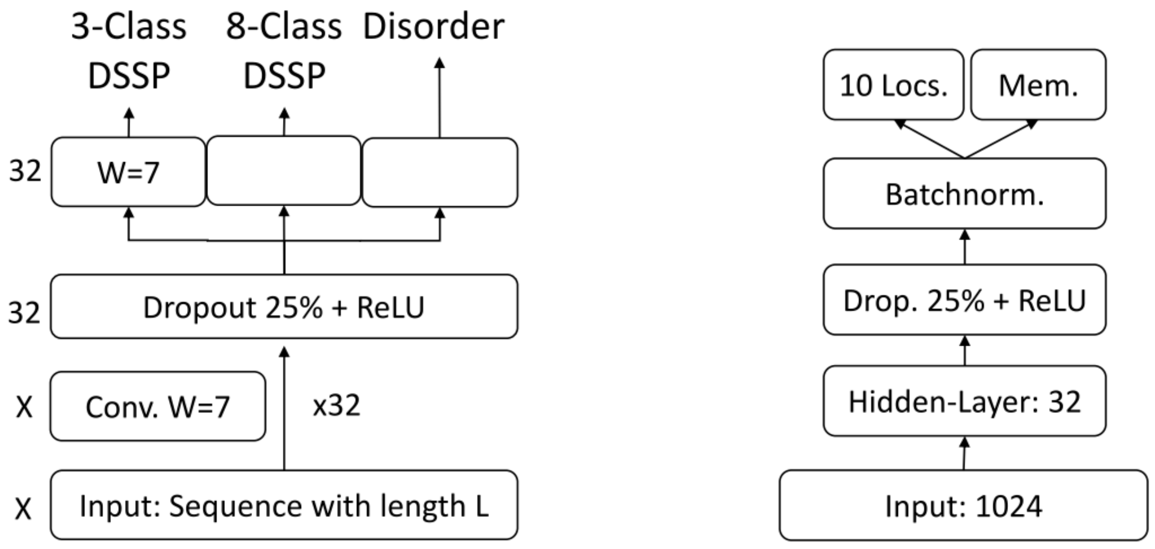


Figure 1: The diagram on the left depicts the architecture trained for residue-level predictions. The 'X' indicates the difference in number of input channels, depending on the type of input feature, e.g. 1024 for SeqVec or 50 for profile-based input. The 'W' gives the window size of the corresponding convolutional layer (W=7 refers here to convolution of size 7x1). On the right, the architecture used for protein-level prediction is shown.

Evaluation measures. To simplify the comparisons, we copied the evaluation measures from the publications that gave our data sets, i.e. those used to develop *NetSurfP-2.0* [40] and *DeepLoc* [39] as far as possible. All numbers reported constituted averages over all proteins in the final test sets. This work aimed at a proof-of-principle that the *SeqVec* embedding contain predictive information. In the absence of any claim for state-of-the-art performance, we did not calculate any significance values for the reported values.

Per-residue performance: Toward this end, we used the standard three-state per-residue accuracy (Q3=percentage correctly predicted in either helix, strand, other [1]) along with its eight-state analog (Q8). Predictions of intrinsic disorder were evaluated through the

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Matthew's correlation coefficient (MCC [66]) and the False-Positive Rate (FPR) representative for tasks with high class imbalance. For completeness, we also provided the entire confusion matrices for both secondary structure prediction problems (Fig. SOM_2). Standard errors were calculated over the distribution of each performance measure for all proteins.

Per-protein performance: The predictions whether a protein was membrane-bound or water-soluble were evaluated by calculating the two-state per set accuracy (Q2: percentage of proteins correctly predicted), and the MCC. A generalized MCC using the Gorodkin measure [54] for K (=10) categories as well as accuracy (Q10), was used to evaluate localization predictions. Standard errors were calculated using 1000 bootstrap samples, each chosen randomly by selecting a sub-set of the predicted test set which had the same size (draw with replacement).

Results

Per-residue performance high but not top. NetSurfP-2.0 uses HHblits profiles along with advanced combinations of Deep Learning architectures to seemingly have become the best method for protein secondary structure prediction [40] reaching a three-state per-residue accuracy Q3 of 82-85% (lower value: small very non-redundant CASP12 set, upper value: larger slightly more redundant TS115 and CB513 sets; Table 1, Fig. 2). All six applications compared here (DeepProf, DeepSeqVec, DeepProf+SeqVec, DeepProtVec, DeepOneHot, DeepBLOSUM65) remained below (Fig. 2A, Table 1). When comparing methods which use only single protein sequences as input (DeepSeqVec, DeepProtVec, DeepOneHot, DeepBLOSUM65), the proposed SeqVec outperformed others by 5-10 (Q3), 5-13 (Q8) and 0.07-0.12 (MCC) percentage points. In our hands, the evolutionary information (DeepProf with HHblits profiles) remained about 4-6 percentage points below NetSurfP-2.0 (Q3=76-81%, Fig. 2, Table 1). Depending on the test set, using SeqVec embeddings instead of evolutionary information (DeepSeqVec: Fig. 2A, Table 1) remained 2-3 percentage points below that mark (Q3=73-79%, Fig. 2A, Table 1). Using both, evolutionary information and SeqVec embeddings (DeepProf+SeqVec) improved over both, but still did not reach the top (Q3=77-82%). In fact, the embedding alone (DeepSeqVec) did not surpass any of the existing methods using evolutionary information (Fig. 2A).

For the prediction of intrinsic disorder, we observed the same: NetSurfP-2.0 performed best, our implementation of evolutionary information (DeepProf) performed worse (Fig. 2B, Table 1). However, for this task the embeddings alone (DeepSeqVec) performed relatively better numerically exceeding the evolutionary information (DeepSeqVec MCC=0.575-0.591 vs. DeepProf MCC=0.506-0.516, Table 1). The combination of evolutionary information with embeddings (DeepProf+SeqVec) improved over using evolutionary information alone but did not improve over SeqVec. Compared to other methods, the embeddings alone reached similar values (Fig. 2B).

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Table 1: Per-residue predictions: secondary structure and disorder ◊

Data	*Prediction task*	*Secondary structure*		*Disorder*	
	Method	*Q3 (%)*	*Q8 (%)*	*MCC*	*FPR*
CASP12	*NetSurfP-2.0 (hhblits)**	82.4	71.1	0.604	0.011
	*NetSurfP-1.0**	70.9	-	-	-
	*Spider3**	79.1	-	0.582	0.026
	*RaptorX**	78.6	66.1	0.621	0.045
	*Jpred4**	76.0	-	-	-
	DeepSeqVec	73.1 ± 1.3	61.2 ± 1.6	0.575 ±0.075	0.026 ±0.008
	DeepProf	76.4 ± 2.0	62.7 ± 2.2	0.506 ±0.057	0.022 ±0.009
	DeepProf + SeqVec	76.5 ± 1.5	64.1 ±1.5	0.556 ±0.080	0.022 ±0.008
	DeepProtVec	62.8 ± 1.7	50.5 ± 2.4	0.505 ±0.064	0.016 ±0.006
	DeepOneHot	67.1 ± 1.6	54.2 ± 2.1	0.461 ±0.064	0.012 ±0.005
	DeepBLOSUM65	67.0 ± 1.6	54.5 ± 2.0	0.465 ±0.065	0.012 ±0.005
TS115	*NetSurfP-2.0 (hhblits)**	85.3	74.4	0.663	0.006
	*NetSurfP-1.0**	77.9	-	-	-
	*Spider3**	83.9	-	0.575	0.008
	*RaptorX**	82.2	71.6	0.567	0.027
	*Jpred4**	76.7	-	-	-
	DeepSeqVec	79.1 ±0.8	67.6 ±1.0	0.591 ±0.028	0.012 ±0.001
	DeepProf	81.1 ±0.6	68.3 ±0.9	0.516 ±0.028	0.012 ±0.002
	DeepProf + SeqVec	82.4 ±0.7	70.3 ±1.0	0.585 ±0.029	0.013 ±0.003
	DeepProtVec	66.0 ± 1.0	54.4 ± 1.3	0.470 ±0.028	0.011 ±0.002
	DeepOneHot	70.1 ± 0.8	58.5 ± 1.1	0.476 ±0.028	0.008 ±0.001
	Deep BLOSUM65	70.3 ± 0.8	58.1 ± 1.1	0.488 ±0.029	0.007 ±0.001
CB513	*NetSurfP-2.0 (hhblits)**	85.3	72.0	-	-
	*NetSurfP-1.0**	78.8	-	-	-
	*Spider3**	84.5	-	-	-
	*RaptorX**	82.7	70.6	-	-
	*Jpred4**	77.9	-	-	-
	DeepSeqVec	76.9 ± 0.5	62.5 ± 0.6	-	-
	DeepProf	80.2 ± 0.4	64.9 ± 0.5	-	-
	DeepProf + SeqVec	80.7 ± 0.5	66.0 ± 0.5	-	-
	DeepProtVec	63.5 ± 0.4	48.9 ± 0.5	-	-
	DeepOneHot	67.5 ± 0.4	52.9 ± 0.5	-	-
	DeepBLOSUM65	67.4 ± 0.4	53.0 ± 0.5	-	-

* Performance comparison for 3- and 8-class secondary structure prediction as well as disorder prediction for the CASP12, TS115 and CB513 data sets. Accuracies (Q3, Q10) are given in percentage. Results marked with * are taken from NetSurfP-2.0 [40]; the authors did not provide standard errors. DeepSeqVec, DeepProtVec, DeepOneHot and DeepBLOSUM65 use only information from single protein sequences.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Per-proteins performance closer to top. For the task of predicting subcellular localization in ten classes, DeepLoc [39] appeared top with Q10=78% (Fig. 2C, Table 2). Our simple embeddings model did not reach all top methods (DeepLoc [39] with Q10=78%, Fig. 2C, Table 2). However, using SeqVec embeddings instead of evolutionary information, it improved over several methods that use evolutionary information by up to Q10=13%.

Table 2: Per-protein predictions: localization and membrane/globular ◊

	Localization			*Membrane/globular*	
Method	*Q10 (%)*	*Gorodkin (MCC)*		*Q2*	*MCC*
*LocTree2**	61	0.53			
*MultiLoc2**	56	0.49			
*SherLoc2**	58	0.51			
*YLoc**	61	0.53			
*CELLO**	55	0.45			
*iLoc-Euk**	68	0.64			
*WoLF PSORT**	57	0.48			
*DeepLoc**	78	0.73		92.3	0.844
SeqVec	68 ± 1	0.61 ± 0.01		86.8 ± 1.0	0.725 ± 0.021
ProtVec	42 ± 1	0.19 ± 0.01		77.6 ± 1.3	0.531 ± 0.026

◊ Performance for per-protein prediction of subcellular localization and the classification of proteins into membrane-bound and water-soluble. Results marked with * were taken from DeepLoc [39]; the authors did not provide standard errors. The results reported for DeepLoc-BL62 (DeepLoc using BLOSUM62 as input), SeqVec and ProtVec were based on single protein sequences, i.e. methods NOT using evolutionary information (neither during training nor testing).

Performance for the classification into membrane-bound and water-soluble proteins followed a similar trend (Fig. 2D, Table 2): while DeepLoc still performed best (Q2=92.3, MCC=0.844), SeqVec reached just a few percentage points lower (Q2=86.8±1.0, MCC=0.725±0.021; full confusion matrix Figure SOM_3). In contrast to this, another method using only single sequences, ProtVec performed substantially worse (Q2=77.6±1.3, MCC=0.531±0.026).

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

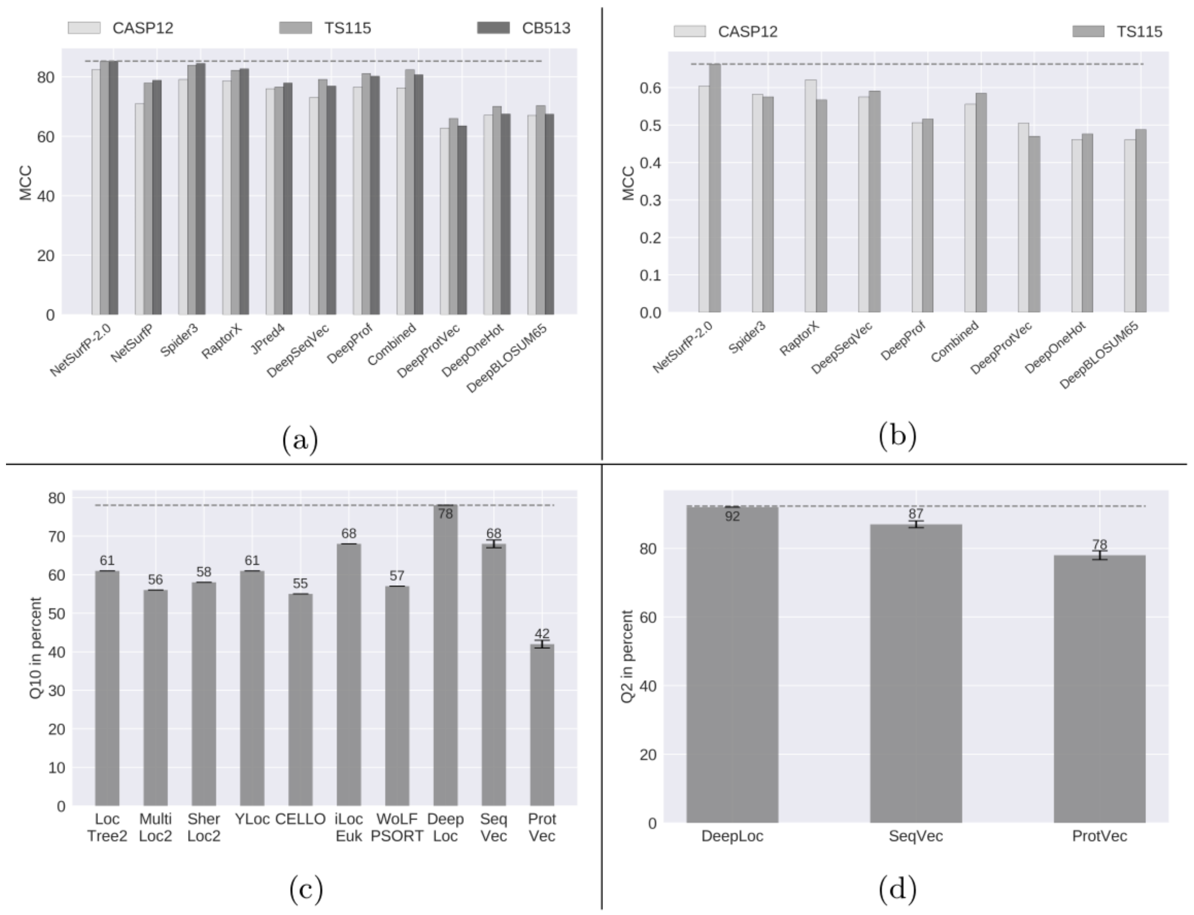
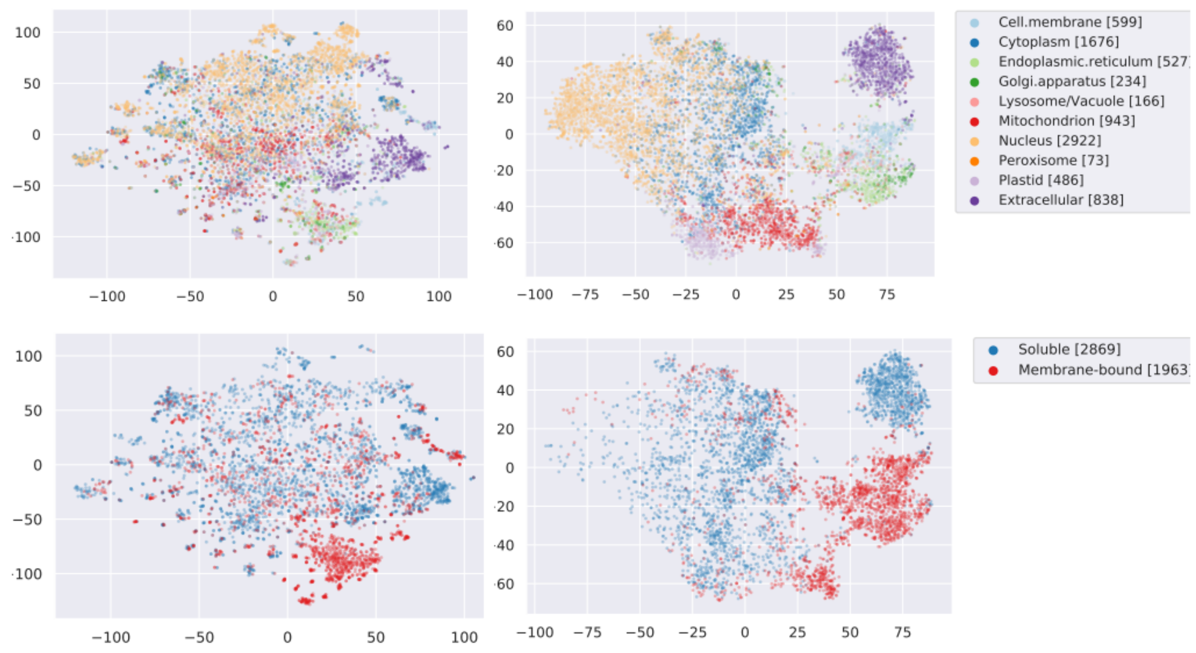


Figure 2: Performance comparisons. The predictive power of the ELMo-like SeqVec embeddings was assessed for per-residue (upper row) and per-protein (lower row). Panel A compared three-state secondary structure prediction of the proposed SeqVec to other encodings based on single protein sequences. Panel B compared predictions of intrinsically disordered residues. Panel C compared per-protein predictions for subcellular localization between top methods (numbers taken from DeepLoc [39] and encodings based on single sequences (ProtVec [64] and our SeqVec). Panel D: the same dataset was used to assess the predictive power of SeqVec for the classification of a protein into membrane-bound and water-soluble.

Visualizing results. Lack of insight often triggers the misunderstanding of machine learning methods as black box solutions barring understanding. In order to interpret the SeqVec embeddings, we have projected the protein-embeddings of the per-protein prediction data upon two dimensions using t-SNE [55]. We performed this analysis once for the raw embeddings (SeqVec, Fig. 3A) and once for the hidden layer representation of our per-protein network after training (Fig. 3B). All t-SNE representations were created using 2,000 iterations and the cosine distance as metric. The two analyses differed only in that the perplexity was set to 25 for one (*SeqVec*) and 50 for the other (hidden layer representation of trained model). The t-SNE representations were colored either according to their localization within the cell (upper row of Fig. 3) or according to whether they are membrane-bound or water-soluble (lower row).

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.



(a) Raw embeddings (SeqVec)

(b) Hidden layer representations after training

Figure 3: t-SNE representations of SeqVec. t-SNE representations of a) raw ELMo embeddings, averaged over all residues in a protein (1024 dimensional; left column), and b) hidden layer representations (32-dimensional, right column) of our per-protein prediction network after training. The redundancy reduced DeepLoc data set was used here. The proteins were colored according to their subcellular localization annotation (upper row) or whether they are membrane-bound or soluble (lower row). The number of samples in each class is given in square brackets to highlight class imbalance. Not all proteins in the DeepLoc set have an annotation for the classification into membrane-bound and soluble which is why the total number of proteins in both sets differ.

Despite never provided during training, the raw embeddings appeared to capture some signal for classifying proteins by localization (Fig. 3A, upper row). The most consistent signal was provided by extra-cellular proteins. Proteins attached to the cell membrane or located in the *endoplasmic reticulum* also formed well-defined clusters. In contrast, the raw embeddings neither captured an ambiguous signal for nuclear nor for mitochondrial proteins. Through training, the network improved the signal to reliably classify mitochondrial and plastid proteins. However, proteins in nucleus and cell membrane continued to be poorly distinguished by t-SNE.

Coloring the t-SNE representations for membrane-bound or water-soluble proteins (Fig. 3, lower row), revealed that the raw embeddings already provided well-defined clusters although never trained on membrane prediction (Fig. 3A). After training, the simple classification was even better (Fig. 3B).

CPU/GPU Time used. Due to the sequential nature of LSTMs, the time required to embed a protein grew linearly with protein length. Depending on the available main memory or GPU memory, this process could be massively parallelized. To optimally use available memory, batches are typically based on tokens rather than on sentences. Here, we sorted proteins

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

according to their length and created batches of ≤15K tokens that could still be handled by a single Nvidia GeForce GTX1080 with 8GB VRAM. On average the processing of a single protein took 0.027s when applying this batch-strategy to the NetSurfP-2.0 dataset (average protein length: 256 residues, i.e. shorter than proteins for which 3D structure is not known). The batch with the shortest proteins (on average 38 residues corresponding to 15% of the total average) required about one tenth (0.003s per protein, i.e. 11% of that for whole set) that for the longest (1578 residues on average corresponding to 610% of the total average), took about six times more (1.5s per protein, i.e. 556% of that for whole set). When creating SeqVec for the DeepLoc set (average length: 558 residues; as this set does not require a 3D structure, it provides a more realistic view on the distribution of protein lengths), the average processing time for a single protein was 0.08 with a minimum of 0.006 for the batch containing the shortest sequences (67 residues on average) and a maximum of 14.5s (9860 residues on average). Roughly, processing time was linear with respect to protein length. On a single Intel i7-6700 CPU with 64GB RAM, processing time increased by roughly 50% to 0.41s per protein, with a minimum and a maximum computation time of 0.06 and 15.3s, respectively. Compared to an average processing time of one hour for 1000 proteins when using evolutionary information directly [40], this implied an average speed up of 120-fold on a single GeForce GTX1080 and 9-fold on a single i7-6700 when predicting structural features; the inference time of DeepSeqVec for a single protein is on average 0.0028s.

Discussion

ELMo alone did not suffice for top performance. On the one hand, none of our implementations of ELMo reached anywhere near today's best (NetSurfP-2.0 for secondary structure and protein disorder and DeepLoc for localization and membrane protein classification; Fig. 2, Table 1, Table 2). Clearly, "just" using ELMo did not suffice to crack the challenges. On the other hand, some of our solutions appeared surprisingly competitive given the simplicity of the architectures. In particular, for the per-protein predictions for which SeqVec clearly outperformed the previously popular ProtVec [64] approach and even commonly used expert solutions (Fig. 1, Table 2: no method tested other than the top-of-the-line DeepLoc reached higher numerical values). For that comparison, we used the same data sets but could not rigorously compare standard errors which were unavailable for other methods. Estimating standard errors for our methods, suggested that the differences were statistically significant as the difference was more than 7 sigma for all methods except DeepLoc (Q10=78) and iLoc-Euk(Q10=68). The results for localization prediction implied that frequently used methods using evolutionary information (all marked with stars in Table 2) did not clearly outperform our ELMo-based simplistic tool (SeqVec in Table 2). This was very different for the per-residue prediction tasks: here almost all top methods using evolutionary information numerically outperformed the simple ELMo-based tool (DeepSeqVec in Fig. 2 and Table 1).

When we combined ELMo with evolutionary information for the per-residue predictions, the resulting tool still did not quite achieve top performance (Q3(NetSurfP-2.0)=85.3% vs. Q3(DeepProf + SeqVec)=82.4%, Table 1). This might suggest some limit for the usefulness of ELMo/SeqVec. However, it might also point to the more advanced solutions realized by NetSurfP-2.0 which applies two LSTMs of similar complexity as our entire system (including ELMo) on top of their last step leading to 35M (35 million) free parameters compared to about

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

244K for DeepProf + SeqVec. Twenty times more free parameters might explain some fraction of the success; we could not simply test how much due to limited GPU resources.

Why did the ELMo-based approach improve more (relative to competition) for per-protein than for per-residue predictions? Per-protein data sets were over two orders of magnitude smaller than those for per-residue (simply because every protein constitutes one sample in the first and protein length samples for the 2nd). Possibly, ELMo helped more for the smaller sets because the unlabelled data pre-processed the data so meaningfully that less had to be learned. This view was strongly underlined by the t-SNE [55] results (Fig. 3): ELMo apparently had learned enough to realize a very rough clustering into localization and membrane/not.

We picked four particular tasks as proof-of-principle for our ELMo/SeqVec approach. All were picked because for these recent methods implemented deep learning to push the field forward and made their data readily available. We cannot imagine why our findings should hold for other tasks of protein prediction. We assume that our findings will be more relevant for small data sets than for large ones. For instance, we assume predictions of inter-residue contacts to improve less, and those for protein binding sites possibly more.

Good and fast predictions without using evolutionary information. SeqVec predicted secondary structure and protein disorder over 100-times faster on a single 8GB GPU than the top-of-the-line prediction method NetSurfP-2.0 which requires machines with hundreds of GB main memory to run MMSeqs2 [32]. For some applications, the speedup might outweigh the reduction in performance.

Modeling the language of life? Our ELMo implementation learned to model a probability distribution over a protein sequence. The sum over this probability distribution constituted a very informative input vector for any machine learning task. It also picked up context-dependent protein motifs without explicitly explaining what these motifs are relevant for. In contrast, tools such as ProtVec will always create the same vectors for a k-mer, regardless of the residues surrounding this k-mer in a particular protein sequence.

Our hypothesis had been that the ELMo embeddings learned from large databases of protein sequences (without annotations) could extract a *modeling of the language of life* in the sense that the resulting system will extract aspects relevant both for per-residue and per-protein prediction tasks. All the results presented, have added independent evidence in full support of this hypothesis. For instance, the three state per-residue accuracy for secondary structure prediction improved by over eight percentage points through ELMo (Table 1: e.g. for CB513: Q3(DeepSeqVec)=76.9% vs. Q3(DeepBLOSUM65)=67.4%, i.e. 9.4 percentage points corresponding to 14% of 67.4), the per-residue MCC for protein disorder prediction also rose (Table 1: e.g. TS115: MCC(DeepSeqVec)=0.591 vs. MCC(DeepBLOSUM65)=0.488, corresponding to 18% of 0.488). On the per-protein level, the improvement over the previously popular tool extracting "meaning" from proteins, ProtVec, was even more substantial (Table 1: localization: Q10(SeqVec)=68% vs. Q10(ProtVec)=42%, i.e. 62% rise over 42; membrane: Q2(SeqVec)=86.8% vs. Q2 (ProtVec)=77.6%, i.e. 19% rise over 77.6). We could demonstrate this reality even more directly using the t-SNE [55] results (Fig. 3): some localizations and the classification of membrane/non-membrane had been implicitly learned by SeqVec without any

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

training. Clearly, our ELMo-driven implementation succeeded to model some aspects of the language of life as proxied for proteins.

Conclusion

We have shown that it is possible to capture and transfer knowledge, e.g. biochemical or biophysical properties, from a large unlabeled database of protein sequences to smaller, labelled datasets. In this first proof-of-principle, our comparably simple models have already reached promising performance for a variety of per-residue and per-protein prediction tasks using only single protein sequences as input without any evolutionary information, i.e. without alignments. This reduces the dependence on the time-consuming and computationally intensive calculation of protein profiles, allowing the prediction of a whole proteome within less than an hour. For instance, on a single GPU the creation of embeddings and predictions of secondary structure and subcellular localization for the whole human proteome took about 32 minutes. Building more sophisticated architectures on top of the proposed SeqVec will increase sequence-based performance further.

Our new SeqVec embeddings may constitute an ideal starting point for many different applications in particular when labelled data are limited. The embeddings combined with evolutionary information might even improve over the best available methods, i.e. enable high-quality predictions. Alternatively, they might ease high-throughput predictions of whole proteomes when used as the only input feature. Alignment-free predictions bring speed and improvements for proteins for which alignments are not readily available or limited, such as for intrinsically disordered proteins, for the Dark Proteome, or for particular unique inventions of evolution. The trick was to tap into the potential of Deep Learning through transfer learning from large repositories of unlabeled data by modeling the language of life.

Acknowledgements

The authors thank primarily to Tim Karl for invaluable help with hardware and software and to Inga Weise for support with many other aspects of this work. We gratefully acknowledge the support of NVIDIA Corporation with the donation of two Titan GPUs used for this research. Also, we want to thank the LRZ (Leibniz Rechenzentrum) for providing us access to one of their DGXV100. We would like to thank Microsoft Research Munich for providing the Rostlab with the Microsoft Education Grant. This work was supported by a grant from the Alexander von Humboldt foundation through the German Ministry for Research and Education (BMBF: Bundesministerium fuer Bildung und Forschung) as well as by a grant from Deutsche Forschungsgemeinschaft (DFG–GZ: RO1320/4–1). Last, not least, thanks to all those who deposit their experimental data in public databases, and to those who maintain these databases.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

References

References

1. Rost, B. and C. Sander, *Prediction of protein secondary structure at better than 70% accuracy.* Journal of Molecular Biology, 1993. **232**: p. 584-599.
2. Rost, B. and C. Sander, *Improved prediction of protein secondary structure by use of sequence profiles and neural networks.* Proceedings of the National Academy of Sciences, 1993. **90**: p. 7558-7562.
3. Rost, B. and C. Sander, *Jury returns on structure prediction.* Nature, 1992. **360**: p. 540.
4. Rost, B. and C. Sander, *Combining evolutionary information and neural networks to predict protein secondary structure.* Proteins: Structure, Function, and Genetics, 1994. **19**: p. 55-72.
5. Solovyev, V.V. and A.A. Salamov, *Predicting a-helix and b-strand segments of globular proteins.* Computer Applications in Biological Science, 1994. **10**: p. 661-669.
6. Barton, G.J., *Protein secondary structure prediction.* Current Opinion in Structural Biology, 1995. **5**: p. 372-376.
7. Chandonia, J.-M. and M. Karplus, *Neural networks for secondary structure and structural class predictions.* Protein Science, 1995. **4**: p. 275-285.
8. Mehta, P.K., J. Heringa, and P. Argos, *A simple and fast approach to prediction of protein secondary structure from multiply aligned sequences with accuracy above 70%.* Protein Science, 1995. **4**: p. 2517-2525.
9. Frishman, D. and P. Argos, *Knowledge-based protein secondary structure assignment.* Proteins: Structure, Function, and Genetics, 1995. **23**: p. 566-579.
10. Jones, D.T., *Protein secondary structure prediction based on position-specific scoring matrices.* Journal of Molecular Biology, 1999. **292**(2): p. 195-202.
11. Rost, B., et al., *Transmembrane helix prediction at 95% accuracy.* Protein Science, 1995. **4**: p. 521-533.
12. Rost, B., R. Casadio, and P. Fariselli, *Topology prediction for helical transmembrane proteins at 86% accuracy.* Protein Science, 1996. **5**: p. 1704-1718.
13. Bigelow, H., et al., *Predicting transmembrane beta-barrels in proteomes.* Nucleic Acids Research, 2004. **32**: p. 2566-2577.
14. Rost, B. and C. Sander, *Conservation and prediction of solvent accessibility in protein families.* Proteins: Structure, Function, and Genetics, 1994. **20**(3): p. 216-226.
15. Radivojac, P., et al., *Protein flexibility and intrinsic disorder.* Protein Science, 2004. **13**: p. 71-80.
16. Schlessinger, A. and B. Rost, *Protein flexibility and rigidity predicted from sequence.* Proteins: Structure, Function, and Bioinformatics, 2005. **61**(1): p. 115-126.
17. Punta, M. and B. Rost, *PROFcon: novel prediction of long-range contacts.* Bioinformatics, 2005. **21**(13): p. 2960-2968.
18. Peng, K., et al., *Optimizing long intrinsic disorder predictors with protein evolutionary information.* J Bioinform Comput Biol, 2005. **3**(1): p. 35-60.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

19. Schlessinger, A., J. Liu, and B. Rost, *Natively unstructured loops differ from other loops.* PLoS Computational Biology, 2007. **3**(7): p. e140.

20. Schlessinger, A., M. Punta, and B. Rost, *Natively unstructured regions in proteins identified from contact predictions.* Bioinformatics, 2007. **23**(18): p. 2376-2384.

21. Nair, R. and B. Rost, *Better prediction of sub-cellular localization by combining evolutionary and structural information.* Proteins: Structure, Function, and Bioinformatics, 2003. **53**(4): p. 917-930.

22. Nair, R. and B. Rost, *Mimicking cellular sorting improves prediction of subcellular localization.* Journal of Molecular Biology, 2005. **348**(1): p. 85-100.

23. Ofran, Y. and B. Rost, *Protein-protein interaction hot spots carved into sequences.* PLoS Computational Biology, 2007. **3**(7): p. e119.

24. Ofran, Y. and B. Rost, *ISIS: Interaction Sites Identified from Sequence.* Bioinformatics, 2007. **23**(2): p. e13-16.

25. Marino Buslje, C., et al., *Networks of high mutual information define the structural proximity of catalytic sites: implications for catalytic residue identification.* PLoS Comput Biol, 2010. **6**(11): p. e1000978.

26. Bromberg, Y. and B. Rost, *SNAP: predict effect of non-synonymous polymorphisms on function.* Nucleic Acids Research, 2007. **35**(11): p. 3823-3835.

27. Adzhubei, I.A., et al., *A method and server for predicting damaging missense mutations.* Nature methods, 2010. **7**(4): p. 248-9.

28. Suzek, B.E., et al., *UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches.* Bioinformatics, 2015. **31**(6): p. 926-32.

29. Altschul, S.F., et al., *Gapped Blast and PSI-Blast: a new generation of protein database search programs.* Nucleic Acids Research, 1997. **25**: p. 3389-3402.

30. Remmert, M., et al., *HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment.* Nat Methods, 2012. **9**(2): p. 173-5.

31. Steinegger, M., et al., *HH-suite3 for fast remote homology detection and deep protein annotation.* bioRxiv, 2019: p. 560029.

32. Steinegger, M. and J. Söding, *MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets.* Nature biotechnology, 2017. **35**(11): p. 1026.

33. Uversky, V.N., et al., *Prediction of intrinsic disorder and its use in functional proteomics.* Methods Mol Biol, 2007. **408**: p. 69-92.

34. Dunker, A.K., et al., *What's in a name? Why these proteins are intrinsically disordered: Why these proteins are intrinsically disordered.* Intrinsically Disord Proteins, 2013. **1**(1): p. e24157.

35. Perdigao, N., et al., *Unexpected features of the dark proteome.* Proc Natl Acad Sci U S A, 2015.

36. Peters, M.E., et al., *Deep contextualized word representations.* arXiv, 2018. **arXiv:1802.05365**.

37. Hochreiter, S. and J. Schmidhuber, *Long short-term memory.* Neural Computation, 1997. **9**(8): p. 1735-1780.

38. Kabsch, W. and C. Sander, *Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features.* Biopolymers, 1983. **22**: p. 2577-2637.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

39. Almagro Armenteros, J.J., et al., *DeepLoc: prediction of protein subcellular localization using deep learning.* Bioinformatics, 2017. **33**(24): p. 4049.

40. Klausen, M.S., et al., *NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning.* Proteins, 2019.

41. Cuff, J.A. and G.J. Barton, *Evaluation and improvement of multiple sequence methods for protein secondary structure prediction.* Proteins: Structure, Function, and Genetics, 1999. **34**(4): p. 508-519.

42. Goldberg, T., T. Hamp, and B. Rost, *LocTree2 predicts localization for all domains of life.* Bioinformatics, 2012. **28**(18): p. i458-i465.

43. Blum, T., S. Briesemeister, and O. Kohlbacher, *MultiLoc2: integrating phylogeny and Gene Ontology terms improves subcellular protein localization prediction.* BMC Bioinformatics, 2009. **10**: p. 274.

44. Briesemeister, S., et al., *SherLoc2: a high-accuracy hybrid method for predicting subcellular localization of proteins.* J Proteome Res, 2009. **8**(11): p. 5363-6.

45. Yu, C.S., et al., *Prediction of protein subcellular localization.* Proteins, 2006. **64**(3): p. 643-51.

46. Chou, K.C., Z.C. Wu, and X. Xiao, *iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins.* PLoS One, 2011. **6**(3): p. e18258.

47. Horton, P., et al., *WoLF PSORT: protein localization predictor.* Nucleic Acids Res, 2007. **35**(Web Server issue): p. W585-7.

48. Briesemeister, S., J. Rahnenfuhrer, and O. Kohlbacher, *YLoc - an interpretable web server for predicting subcellular localization.* Nucleic Acids Res, 2010. **38 Suppl**: p. W497-502.

49. Boutet, E., et al., *UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View.* Methods Mol Biol, 2016. **1374**: p. 23-54.

50. Li, W. and A. Godzik, *Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences.* Bioinformatics, 2006. **22**(13): p. 1658-9.

51. Fu, L., et al., *CD-HIT: accelerated for clustering the next-generation sequencing data.* Bioinformatics, 2012. **28**(23): p. 3150-3152.

52. Moussa, M. and Mandoiu, II, *Single cell RNA-seq data clustering using TF-IDF based methods.* BMC Genomics, 2018. **19**(Suppl 6): p. 569.

53. Bailey, T.L., et al., *MEME SUITE: tools for motif discovery and searching.* Nucleic Acids Res, 2009. **37**(Web Server issue): p. W202-8.

54. Bernard, G., C.X. Chan, and M.A. Ragan, *Alignment-free microbial phylogenomics under scenarios of sequence divergence, genome rearrangement and lateral genetic transfer.* Sci Rep, 2016. **6**: p. 28970.

55. Nakai, K. and P. Horton, *PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization.* Trends Biochem Sci, 1999. **24**(1): p. 34-6.

56. Kuang, R., et al., *Profile-based string kernels for remote homology detection and motif extraction.* J Bioinform Comput Biol, 2005. **3**(3): p. 527-50.

57. Leslie, C., et al., *Mismatch string kernels for SVM protein classification.* Bioinformatics, 2003: p. in press.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted April 19, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

58. Noble, W.S., et al., *Identifying remote protein homologs by network propagation.* FEBS J, 2005. **272**(20): p. 5119-28.

59. Hamp, T. and B. Rost, *Evolutionary profiles improve protein-protein interaction prediction from sequence.* Bioinformatics, 2015. **31**(12): p. 1945-50.

60. Mikolov, T., et al., *Efficient estimation of word representations in vector space.* ArXiv, 2013: p. arXiv:1301.3781.

61. Asgari, E., A.C. McHardy, and M.R.K. Mofrad, *Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX).* Sci Rep, 2019. **9**(1): p. 3577.

62. Bojanowski, P., et al., *Enriching word vectors with subword information.* Transactions of the Association for Computational Linguistics, 2017. **5**: p. 135-146.

63. Pennington, J., R. Socher, and C. Manning. *Glove: Global vectors for word representation*. in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

64. Asgari, E. and M.R. Mofrad, *Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics.* PLoS One, 2015. **10**(11): p. e0141287.

65. Henikoff, S. and J.G. Henikoff, *Amino acid substitution matrices from protein blocks.* Proceedings of the National Academy of Sciences, 1992. **89**: p. 10915-10919.

66. Matthews, B.W., *Comparison of the predicted and observed secondary structure of T4 phage lysozyme.* Biochimica et Biophysica Acta, 1975. **405**: p. 442-451.