bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

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 Nechaev [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 [at] rostlab.org, tel: +49-289-17-811 (email rost: assistant [@] rostlab.org)

Abstract

Background: 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. The retrieval of related proteins from ever growing sequence databases is becoming so time-consuming that the analysis of entire proteomes becomes challenging. On top, evolutionary information is less powerful for small families, e.g. for proteins from the *Dark Proteome*.

Results: We introduce a novel way to represent protein sequences as continuous vectors (*embeddings*) by using the deep bi-directional model ELMo taken from natural language processing (NLP). The model has effectively captured the biophysical properties of protein sequences from unlabeled big data (UniRef50). After training, this knowledge is transferred to single protein sequences by predicting relevant sequence features. We refer to these new embeddings as *SeqVec* (*Seq*uence-to-*Vec*tor) and demonstrate their effectiveness by training simple convolutional neural networks on existing data sets for two completely different prediction tasks. At the per-residue level, we significantly improved secondary structure (for NetSurfP-2.0 data set: Q3=79%±1, Q8=68%±1) and disorder predictions (MCC=0.59±0.03) over methods not using evolutionary information. At the per-protein level, we predicted subcellular localization in ten classes (for DeepLoc data set: Q10=68%±1) and distinguished membrane-bound from water-soluble proteins (Q2= 87%±1). All results built upon the embeddings gained from the new tool *SeqVec* neither explicitly nor implicitly using evolutionary information. Nevertheless, it improved over some methods using such information. Where the lightning-fast *HHblits* needed on average about two minutes to generate the evolutionary information for a target protein, *SeqVec* created the vector representation on average in 0.03 seconds.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Conclusion: We have shown that transfer learning can be used to capture biochemical or biophysical properties of protein sequences from large unlabeled sequence databases. The effectiveness of the proposed approach was showcased for different prediction tasks using only single protein sequences. *SeqVec* embeddings enable predictions that outperform even some methods using evolutionary information. Thus, they prove to condense the underlying principles of protein sequences. This might be the first step towards competitive predictions based only on single protein sequences.

Availability: SeqVec: https://github.com/mheinzinger/SeqVec Prediction server: https://embed.protein.properties

Key words: Machine Learning, Language Modeling, Sequence Embedding, Secondary structure prediction, Localization prediction, Transfer Learning, Deep Learning.

Abbreviations: 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;

Background

Over two decades ago, the combination of evolutionary information (from Multiple Sequence Alignments – MSA) and machine learning (standard feed-forward artificial neural networks – ANN) completely changed protein secondary structure prediction [1-3]. The concept was quickly taken up [4-8], and it was shown how the improvement in prediction increased with even larger families including more diverse evolutionary information [9, 10]. The idea was applied to other tasks, including the prediction of transmembrane regions [11-13], solvent accessibility [14], residue flexibility (B-values) [15, 16], inter-residue contacts [17] and protein disorder [15, 18-20]. Later, methods predicting aspects of protein function improved through the combination of evolutionary information and machine learning, 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]. Arguably, the most important breakthrough for protein structure prediction over the decade was a more efficient way of using evolutionary couplings [28-31].

Although evolutionary information has become increasingly crucial, it is also becoming increasingly costly. Firstly, bio-sequence databases grow faster than computers making it expensive to find and align related proteins. For instance, the number of UniProt entries is now more than doubling every two years [32]. Consequently, methods as fast as PSI-BLAST [33] have to be replaced by faster solutions such as HHblits [34]. Even its latest version HHblits3 [35] still needs several minutes to search UniRef50 (subset of UniProt; ~20% of UniProt release 2019_02). The next step up in speed such as MMSeqs2 [36] appear to cope with the challenge at the expense of increasing hardware requirements while databases keep growing. However, even these solutions might eventually lose the battle against the speedup of sequencing. Analyzing data sets involving millions of proteins, i.e. samples of the human gut microbiota or metagenomic samples, will be one of the major challenges for computational biology. Secondly, evolutionary information is missing for some proteins, e.g. for proteins with substantial intrinsically disordered regions [15, 37, 38], or the entire *Dark Proteome* [39] full of proteins that are less-well studied but important for function [40].

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Here, we propose a novel embedding of protein sequences that replaces the explicit search for evolutionary related proteins by an implicit transfer of biophysical information derived from large, unlabeled sequence data (here UniRef50). Towards this end, we have adopted a method that has been revolutionizing Natural Language Processing (NLP), namely the bi-directional language model ELMo (Embeddings from Language Models) [41]. In NLP, ELMo is trained on unlabeled text-corpora 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 a given word depend on its context. This has the advantage that two identical words can have different embeddings, depending on the words surrounding them.

We hypothesized that the ELMo concept could be applied to learn aspects of what makes up the language of life distilled in protein sequences. Three main challenges arose. (1) Proteins range from about 30 to 33,000 residues, a much larger range than for the average English sentence extending over 15-30 words [42], and even more extreme than notable literary exceptions such as James Joyce's Ulysses (1922) with almost 4,000 words in a sentence. Longer proteins require more GPU memory and the underlying models (so called Long Short-Term Memory networks (LSTMs) [43]) have only a limited capability to remember long-range dependencies. (2) Proteins mostly use 20 standard amino acids, 100,000 times less than in the English language. 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 amino acids) than the largest existing NLP corpus (1 billion words). Simply put: Wikipedia is roughly ten times larger than Webster's Third New International Dictionary and the entire UniProt is over ten times larger than Wikipedia. As a result, larger models might be required to absorb the information provided.

We trained the bi-directional language model ELMo on UniRef50. Then we assessed the predictive power of the embeddings by application to tasks on two levels: per-residue (word-level) and per-protein (sentence-level). For the per-residue prediction task, we predicted secondary structure in three (helix, strand, other) and eight states (all DSSP [44]), as well as long intrinsic disorder in two states. For the per-protein prediction task, we implemented the predictions of protein subcellular localization in ten classes and a binary classification into membrane-bound and water-soluble proteins. We used publicly available data sets from two recent methods that achieved break-through performance through Deep Learning, namely NetSurfP-2.0 (secondary structure [45]) and DeepLoc (localization [46]). We compared the performance of the *SeqVec* embeddings to state-of-the-art methods, and also to a popular embedding tool for protein sequences, namely *ProtVec* [47]. ProtVec assumes that every token or word consists of three consecutive residues (amino acid 3-mers). During training, each protein sequence in *SwissProt* [48] is split into overlapping 3-mers and another tool from NLP, namely the *word2vec* (skip-gram) model [49] is used to predict adjacent words, given the word at the center. After training, protein sequences can be split into overlapping 3-mers which are mapped onto a 100-dimensional latent space. While this approach can capture local information, it loses information on sequence ordering and the resulting embeddings are insensitive to their context (non-contextualized), i.e. the same word results in the same embedding regardless of the specific context.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Results

***SeqVec* embeddings compress UniProt50.** *SeqVec*, our ELMo-like implementation was trained for three weeks on 5 Nvidia Titan GPUs with 12 GB memory each. The model was trained until its *perplexity* (uncertainty when predicting the next token) converged at around 10.5 (Fig. SOM_1). Training and testing were not split due to technical limitations (incl. shortage of CPU/GPU). ELMo was designed to reduce the risk of overfitting by sharing weights between forward and backward LSTMs and by using dropout. The model had about 93M (mega/million) free parameters which was about 100-times smaller than the 9.6G (giga/billion) tokens to predict. Similar approaches have shown that even todays largest models (750M free parameters), are not able to overfit on a large corpus (250M protein sequences) [50].

Per-residue performance high, not highest. NetSurfP-2.0 uses HHblits profiles along with advanced combinations of Deep Learning architectures [45]. NetSurfP-2.0 could be today's best method for protein secondary structure prediction, 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. 1; note that several contenders such as *Spider3* and *RaptorX* appear to differ by fewer than three standard errors). All six methods developed by us (DeepProf, DeepSeqVec, DeepProf+SeqVec, DeepProtVec, DeepOneHot, DeepBLOSUM65) fell short of reaching this mark (Fig. 1A, Table 1). When comparing methods that use only single protein sequences as input (DeepSeqVec, DeepProtVec, DeepOneHot, DeepBLOSUM65; all white in Table 1), the proposed *SeqVec* outperformed others by 5-10 (Q3), 5-13 (Q8) and 0.07-0.12 (MCC) percentage points. The evolutionary information (DeepProf with HHblits profiles) remained about 4-6 percentage points below NetSurfP-2.0 (Q3=76-81%, Fig. 1, Table 1). Depending on the test set, using *SeqVec* embeddings instead of evolutionary information (DeepSeqVec: Fig. 1A, Table 1) remained 2-3 percentage points below that mark (Q3=73-79%, Fig. 1A, 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. 1A).

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

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

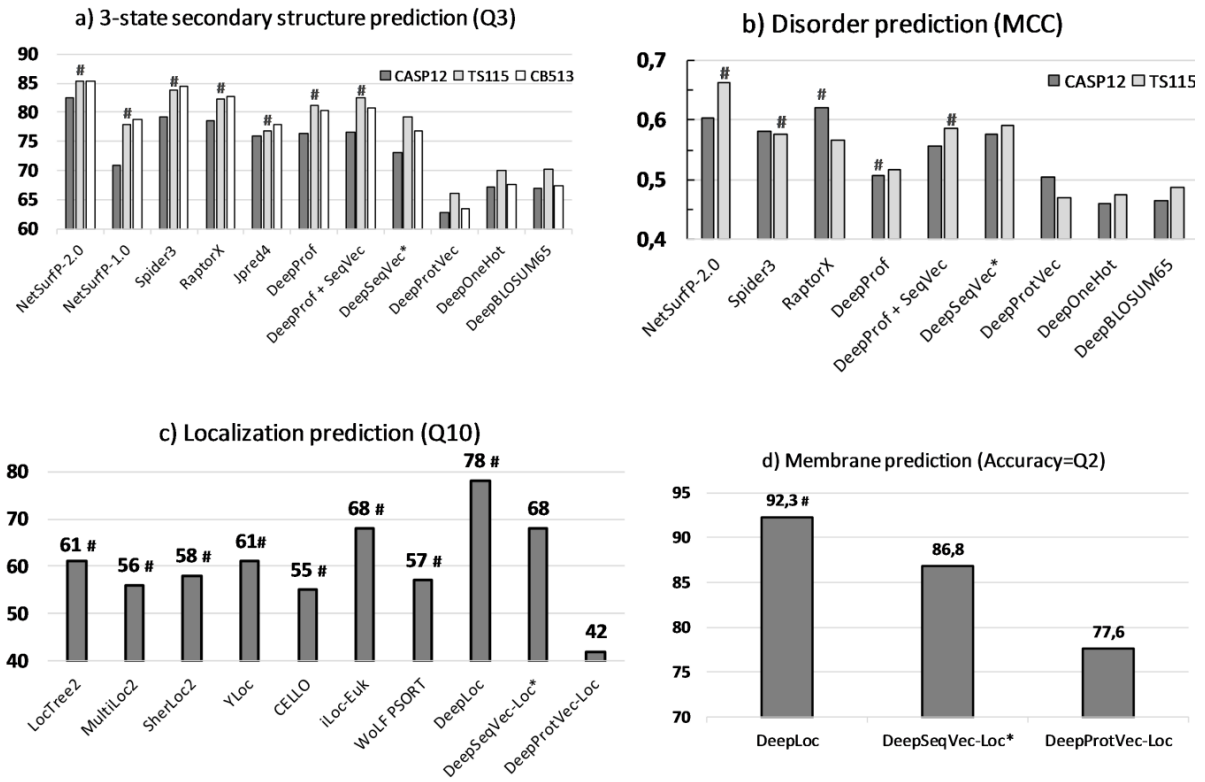


Figure 1: Performance comparisons. The predictive power of the ELMo-like SeqVec embeddings was assessed for per-residue (upper row) and per-protein (lower row). Methods using evolutionary information (mostly in the form of alignments) are highlighted by a '#' above the bar(s) of the methods. Approaches using only the proposed SeqVec embeddings are highlighted by a '*'. Panel A compared three-state secondary structure prediction of the proposed SeqVec to other embeddings 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 [46] and embeddings based on single sequences (ProtVec [47] and our SeqVec). Panel D: the same data set was used to assess the predictive power of SeqVec for the classification of a protein into membrane-bound and water-soluble.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

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 [45]; the authors did not provide standard errors. DeepSeqVec, DeepProtVec, DeepOneHot and DeepBLOSUM65 use only information from single protein sequences. All methods using evolutionary information are marked by gray boxes, these performed best (best in each set marked by bold numbers) for all tasks and measures.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Per-protein performance close to best. For the task of predicting subcellular localization in ten classes, DeepLoc [46] appears to be today's best tool with Q10=78% (Fig. 1C, Table 2). Our simple embeddings model DeepSeqVec-Loc reached second best performance together with iLoc-Euk [51] with Q10=68% (Fig. 1C, Table 2). For this application the SeqVec embeddings outperformed several other 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			
*CELLO**	55	0.45			
*WoLF PSORT**	57	0.48			
*YLoc**	61	0.53			
*SherLoc2**	58	0.51			
*iLoc-Euk**	68	0.64			
*DeepLoc**	**78**	**0.73**		**92.3**	**0.844**
DeepSeqVec-Loc	68 ± 1	0.61 ± 0.01		86.8 ± 1.0	0.725 ± 0.021
DeepProtVec-Loc	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 [46]; the authors did not provide standard errors. The results reported for SeqVec and ProtVec were based on single protein sequences, i.e. methods NOT using evolutionary information (neither during training nor testing). All methods using evolutionary information are marked by gray boxes, these performed best (best in each set marked by bold numbers) for all tasks and measures.

Performance for the classification into membrane-bound and water-soluble proteins followed a similar trend (Fig. 1D, Table 2): while DeepLoc still performed best (Q2=92.3, MCC=0.844), DeepSeqVec-Loc reached just a few percentage points lower (Q2=86.8±1.0, MCC=0.725±0.021; full confusion matrix Figure SOM_2). In contrast to this, ProtVec, another method using only single sequences, performed substantially worse (Q2=77.6±1.3, MCC=0.531±0.026).

Visualizing results. Lack of insight often triggers the misunderstanding that machine learning methods are 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 [52]. We performed this analysis once for the raw embeddings (SeqVec, Fig. 2 upper row) and once for the hidden layer representation of the per-protein network (DeepSeqVec-Loc) after training (Fig. 2 lower row). All t-SNE representations in Fig. 2 were created using 3,000 iterations and the cosine distance as metric. The two analyses differed only in that the perplexity was set to 20 for one (*SeqVec*) and 15 for the other (DeepSeqVec-Loc). The t-SNE representations were colored either according to their

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

localization within the cell (left column of Fig. 2) or according to whether they are membrane-bound or water-soluble (right column).

Despite never provided during training, the raw embeddings appeared to capture some signal for classifying proteins by localization (Fig. 2, upper row, left column). The most consistent signal was visible for 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 the nucleus and cell membrane continued to be poorly distinguished via t-SNE.

Coloring the t-SNE representations for membrane-bound or water-soluble proteins (Fig. 2, right column), revealed that the raw embeddings already provided well-defined clusters although never trained on membrane prediction (Fig. 2, upper row). After training, the classification was even better (Fig. 2, lower row).

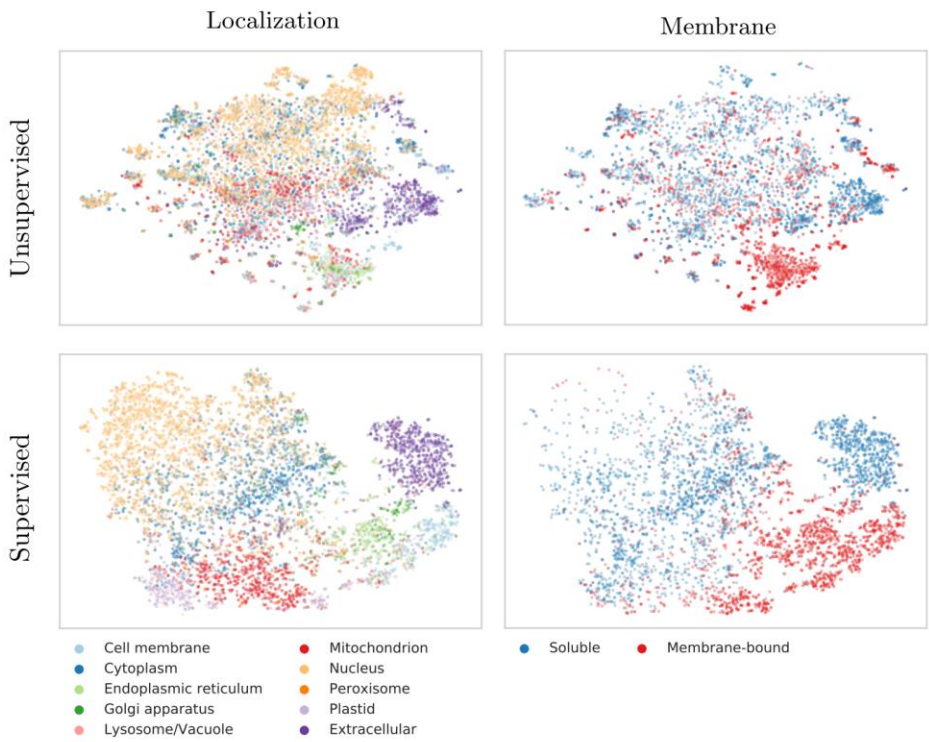


Figure 2: t-SNE representations of SeqVec. t-SNE projections from embedded space onto a 2D representation; upper row: unsupervised 1024-dimensional ELMo embeddings, averaged over all residues in a protein; lower row: supervised 32 dimensional ELMo embeddings, reduced via per-protein machine learning predictions. The redundancy reduced DeepLoc data set was used for this figure. Proteins were colored according to their localization (left column) or whether they are membrane-bound or water-soluble (right column). Not all proteins in the data set have annotations for the classification into membrane-bound/water-soluble and have thus been excluded in the panels on the right.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

The plots in the upper row suggest that the SeqVec embeddings already capture various aspects of proteins prior even to supervised learning. After supervised training (lower row), this information is transferred to, and further distilled by networks with simple architectures to distinguish aspects of protein function (sometimes drastically, as in the case of membrane boundness, as suggested by the coloured clusters in the bottom right panel).

After training machine learning devices (lower row) on the supervised tasks of localization (left) and membrane boundness (right) prediction, the power of SeqVeq embeddings to distinguish aspects of function and structure become even more pronounced, sometimes drastically so, as suggested by the almost fully separable clusters in the lower right panel.

Analogously, we used t-SNE projections to analyze SeqVec embeddings on different levels of complexity inherent to proteins (Fig. 3), ranging from the building blocks (amino acids, Fig. 3a), to secondary structure defined protein classes (Fig. 3b), over functional features (Fig. 3c), and onto the macroscopic level of the kingdoms of life and viruses (Fig. 3d; classifications in panels 3b-3d based on SCOPe [53]). Similar to the results described in [50], our projection of the embedding space confirmed that the model successfully captured bio-chemical and bio-physical properties on the most fine-grained level, i.e. the 20 standard amino acids (Fig. 3a). For example, aromatic amino acids (W, F, Y) are well separated from aliphatic amino acids (A, I, L, M, V) and small amino acids (A, C, G, P, S, T) are well separated from large ones (F, H, R, W, Y). The projection of the letter indicating an unknown amino acid (X), clustered closest to the amino acids alanine (A) and glycine (G) (data not shown). The two amino acids with the smallest side chains might be least biased towards other biochemical features like charge and they are the 2nd (A) and 4th (G) most frequent amino acids in our training set (Table SOM_1). Rare (O, U) and ambiguous amino acids (Z, B) were removed from the projection as their clustering showed that the model could not learn a reasonable embedding from the very small number of samples.

Crude structural classes as defined in SCOPe (Fig. 3b) were also captured by SeqVec embeddings. Although the embeddings were only trained to predict the next amino acid in a protein sequence, well separated clusters emerged from those embeddings in structure space. Especially, membrane proteins and small proteins formed distinct clusters (note: protein length did not affect the t-SNE grouping). Also, these results indicated that the embeddings captured complex relationships between proteins which are not directly observable from sequence similarity alone as SCOPe was redundancy reduced at 40% sequence identity. Therefore, the new embeddings could complement sequence-based structural classification as it was shown that the sequence similarity does not necessarily lead to structural similarity [54].

To further investigate the clusters which emerged from the SCOPe data set, we colored the same data set based on protein functions (Fig. 3c) and kingdoms (Fig. 3d). This analysis revealed that many of the small, distinct clusters emerged based on protein functions. For example, transferases and hydrolases form many small clusters. When increasing the level of abstraction by coloring the proteins according to their kingdoms, we observed certain clusters which are dominated by e.g. eukaryotes. Comparing the different views captured in panels 3b-3 revealed connections, e.g. that all-beta or small proteins dominate in eukaryotes (compare blue and orange islands in Fig. 3b with the same islands in Fig. 3d – colored blue to mark eukaryotes).

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

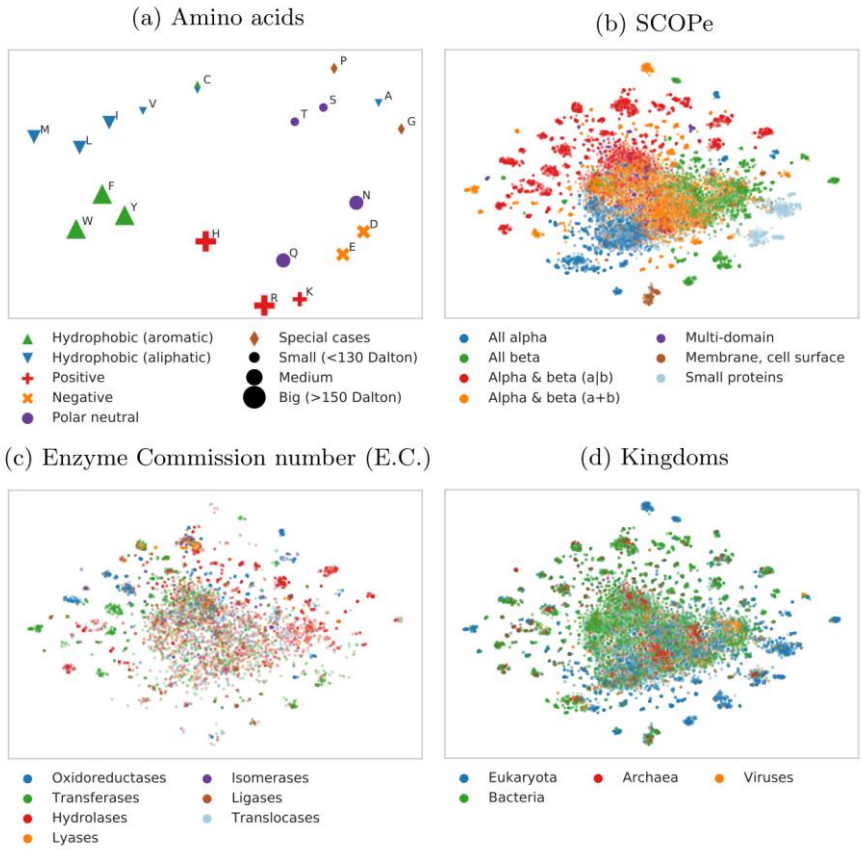


Figure 3: Modeling the language of life. 2D t-SNE projections of unsupervised SeqVec embeddings highlight different realities of proteins and their constituent parts, amino acids. Panels (b) to (d) are based on the same data set (Structural Classification of Proteins – extended (SCOPe) 2.07, redundancy reduced at 40%). For these plots, only subsets of SCOPe containing proteins with the annotation of interest (enzymatic activity (c) and kingdom (d)) may be displayed. Panel (a): the embedding space confirms: the 20 standard amino acids are clustered according to their biochemical and biophysical properties, i.e. hydrophobicity, charge or size. The unique role of Cysteine (C, mostly hydrophobic and polar) is conserved. Panel (b): SeqVec embeddings capture structural information as annotated in the main classes in SCOPe without ever having been explicitly trained on structural features. Panel (c): many small, local clusters share function as given by the main classes in the Enzyme Commission Number (E.C.). Panel (d): similarly, small, local clusters represent different kingdoms of life.

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 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 data set (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 average protein length in the whole data set) required about one tenth (0.003s per protein, i.e.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

11% of that for whole set). The batch containing the longest protein sequences in this data set (1578 residues on average, corresponding to 610% of average protein length in the whole data set), 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 [45], 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. 1, 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 [47] 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 sigmas 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 simple ELMo-based tool (DeepSeqVec-Loc 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 model built on the ELMo embeddings (DeepSeqVec in Fig. 1 and Table 1). However, 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.

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

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

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 second). It is likely that ELMo helped more for the smaller data sets because the unlabeled data is pre-processed so meaningful that less information needs to be learned by the ANN during per-protein prediction. This view was strongly supported by the t-SNE [52] results (Fig. 2): 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. These tasks were picked because recently developed methods implemented deep learning and the associated data sets for training and testing were made publicly available. We cannot imagine why our findings should not hold for other tasks of protein prediction and welcome the community to test our *SeqVec* for their particular tasks. 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 to retrieve evolutionary information summarized in alignments. For some applications, the speedup might outweigh the reduction in performance. Therefore, embedding-based approaches such as *SeqVec* suggested a promising solution toward solving one of the biggest challenges for computational biology: How to efficiently handle the exponentially increasing number of sequences? Here, we showed that relevant information from large unannotated biological databases can be compressed into embeddings that condense and abstract the underlying biophysical principles. These embeddings, essentially the weights of a neural network, help as input to many problems for which smaller sets of annotated data are available (secondary structure, disorder, localization). Although the compression step needed to build the *SeqVec* model is very GPU-intensive, it can be performed in a centralized way using large clusters. After training, the model can be used as input by any consumer hardware. Such solutions are ideal to support researches without access to expensive cluster infrastructure.

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 *probabilistic model 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 here 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

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

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(DeepSeqVec-Loc)=68% vs. Q10(DeepProtVec-Loc)=42%, i.e. 62% rise over 42; membrane: Q2(DeepSeqVec-Loc)=86.8% vs. Q2 (DeepProtVec-Loc)=77.6%, i.e. 19% rise over 77.6). We could demonstrate this reality even more directly using the t-SNE [52] results (Fig. 2 and Fig. 3): different levels of complexity ranging from single amino acids, over some localizations, structural features, functions and the classification of membrane/non-membrane had been implicitly learned by SeqVec without any training on such data. 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 data set of protein sequences to smaller, labelled data sets. 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 obtainable from only single protein sequences as input, that is: without any direct 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 per-residue and per-protein features of a whole proteome within less than an hour. For instance, on a single GeForce GTX 1080, 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.

Methods

Data. UniRef50 training of *SeqVec*: We trained ELMo on UniRef50 [32], 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

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

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* (*Seq*uence-to-*Vec*tor).

Visualization of projection space: The current release of the "Structural Classification Of Proteins" (SCOPe, [53]) database (2.07) contains 14323 proteins at a redundancy level of 40%. Functions encoded by the Enzyme Commission number (E.C., [55]) were retrieved via the "Structure Integration with Function, Taxonomy and Sequence" (SIFTS) mapping [56]. SIFTS allows among other things a residue-level mapping between UniProt and PDB entries and a mapping from PDB identifiers to E.C.s. If no function annotation was available for a protein or if the same PDB identifier was assigned to multiple E.C.s, it was removed from Fig. 3c. Taxonomic identifiers were used to map proteins to one of the 3 kingdoms of life or to viruses. Again, proteins were removed if no such information was available. The number of iterations for the t-SNE projections was set again to 3000 and the perplexity was adjusted (perplexity=5 for Fig. 3a and perplexity=30 for Fig. 3b-3d).

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* [45]. Performance values for the same data set exist also for other recent methods such as *Spider3* [57], *RaptorX* [58, 59] and *JPred4* [60]. The set contains 10,837 sequence-unique (at 25% PIDE) proteins of experimentally known 3D structures from the PDB [61] with a resolution of 2.5 Å (0.25 nm) or better, collected by the PISCES server [62]. DSSP [44] 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 others to O (other; often misleadingly referred to as *coil* or *loop*). As the authors of NetSurfP-2.0 did not include the raw protein sequences in their public data set, we used the SIFTS file to obtain the original sequence. 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** [63]: 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** [64]: 513 non-redundant sequences compiled 20 years ago (511 after SIFTS mapping); **CASP12** [65]: 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). Each of these sets covers different aspects of the secondary structure prediction problem: CB513 and TS115 only use structures determined by X-ray crystallography, applying similar cutoffs with respect to redundancy (30%) and resolution (2.5-3.0Å). While they serve as a good proxy for a baseline performance, CASP12 might better reflect the true generalization capability for unseen proteins including NMR and Cryo-EM as means of structure determination. Also, the strict redundancy reduction based on publication date reduces the bias towards well studied families. 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

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

compare methods using evolutionary information and those using our new word embeddings, we took the *HHblits* profiles published along with the NetSurfP-2.0 data set.

Per-protein level: localization & membrane proteins (DeepLoc). Localization prediction was trained and evaluated using the *DeepLoc* data set [46] for which performance was measured for several methods, namely: LocTree2 [66], MultiLoc2 [67], SherLoc2 [68], CELLO [69], iLoc-Euk [51], WoLF PSORT [70] and YLoc [71]. The data set contained proteins from UniProtKB/Swiss-Prot [48] (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). The resulting 13,858 proteins were clustered through PSI-CD-HIT [72, 73] (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).

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 [33], clustering [74], motifs [75, 76], and prediction methods [66, 70, 77-81]. 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 in *Word2Vec* [49] and showcased for computational biology through *ProtVec* [49, 82]. More specialized implementations are *mut2vec* [83] learning mutations in cancer, and *phoscontext2vec* [84] identifying phosphorylation sites. The performance of context-insensitive approaches was pushed to its limits by adding sub-word information (FastText [85]) or global statistics on word co-occurance (GloVe [86]). Context-sensitive word embedding started a new wave of word embedding techniques for NLP in 2018: the particular embedding renders the meaning of the phrase "*paper tiger*" dependent upon the context. Popular examples like ELMo [41] and Bert [87] 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, in this work we focused on ELMo as it allows processing of sequences of variable length. This ELMo model consists of a single CharCNN [88] over the characters in a word and two layers of bidirectional LSTMs that introduce the context information of surrounding words. The CharCNN transforms all characters within a single word via an embedding layer into vector space and runs multiple CNNs of varying window size (here: ranging from 1 to 7) and number of filters (here: 32, 64, …, 1024). In order to obtain a fixed-dimensional vector for each word, regardless of its length, the output of the CNNs is max-pooled and concatenated. The first bi-directional LSTM operates directly on the output of the CharCNN, while the second LSTM layer takes the output of the first LSTM as input. As

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

described in the original ELMo publication, the weights of the forward and the backward model are shared in order to reduce the memory overhead of the model and to combat overfitting. Even though, the risk of overfitting is small due to the high imbalance between number of trainable parameters (93M) versus number of tokens (9.3B), dropout at a rate of 10% was used to reduce the risk of overfitting. This model is trained to predict the next word given all previous words in a sentence. To the best of our knowledge, the context-sensitive ELMo has not been adapted to protein sequences, yet.

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. 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 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 (1 CharCNN-layer, 2 LSTM-layers), 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 [41] 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 padded elements ('<MASK>') or the beginning ('<S>') or the end of a sequence ('</S>')), (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.

SeqVec 2 prediction. <u>On the per-residue level</u>, 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 [89] of the ADAM optimizer [90], 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 of 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. These two layers were connected through a rectified linear unit (ReLU) and a dropout layer [91] with a dropout-rate of 25% (Fig. 4, left panel). This simple architecture was trained independently on six different types of input, each with a different number of free parameters. (i) <u>DeepProf (14,000=14k free parameters)</u>: 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 [45], the state transition probabilities of the Hidden-Markov-Model (7 features) and 3 features describing the local alignment diversity. (ii) Deep<u>SeqVec (232k free parameters):</u> Each protein sequence is represented by the output from SeqVec. The resulting embedding described each residue as a 1024-dimensional vector. <u>(iii) DeepProf+SeqVec (244k free parameters):</u> This model simply concatenated the input vectors used in (i) and (ii). <u>(iv) DeepProtVec (25k free parameters):</u> Each sequence was split into overlapping 3-mers each represented by a 100-dimensional ProtVec [47]. <u>(v) DeepOneHot</u>

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

(7k free parameters): The 20 amino acids were encoded as one-hot vectors as described above. Rare amino acids were mapped to vectors 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 [92]. 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 resulting in a 1024-dimensional vector representing any protein in the data set. ProtVec representations were derived the same way, resulting in a 100-dimensional vector. These vectors (either 100- or 1024 dimensional) were first compressed to 32 features, then dropout with a dropout rate of 25%, batch normalization [93] and a rectified linear Unit (ReLU) were applied before the final prediction (Fig. 4, right panel). In the following, we refer to the models trained on the two different input types as (i) DeepSeqVec-Loc (33k free parameters): average over SeqVec embedding of a protein as described above and (ii) DeepProtVec-Loc (320 free parameters): average over ProtVec embedding of a protein. 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 (11 seconds for DeepSeqVec-Loc, 15 minutes for DeepSeqVec).

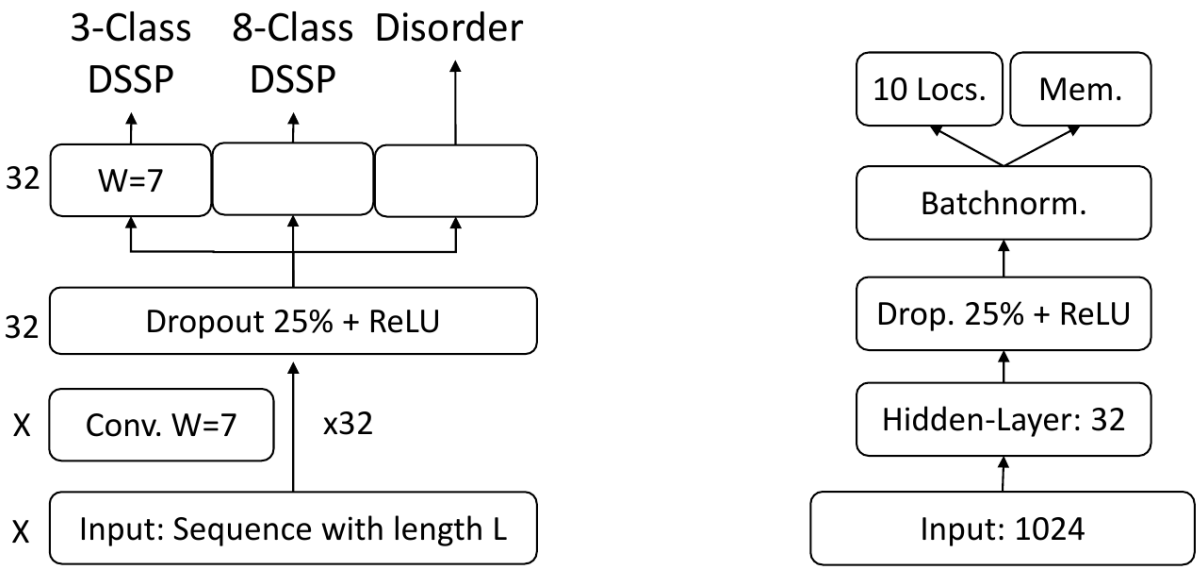


Figure 4: On the left the architecture of the model used for the per-residue level predictions (secondary structure and disorder) is sketched, on the right that used for per-protein level predictions (localization and membrane/not membrane). The 'X', on the left, indicates that different input features corresponded to a difference in the number of input channels, e.g. 1024 for *SeqVec* or 50 for profile-based input. The letter 'W' refers to the window size of the corresponding convolutional layer (W=7 implies a convolution of size 7x1).

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Evaluation measures. To simplify comparisons, we ported the evaluation measures from the publications we derived our data sets from, i.e. those used to develop *NetSurfP-2.0* [45] and *DeepLoc* [46]. 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 Matthew's correlation coefficient (MCC [94]) 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 [95] 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 that had the same size (draw with replacement).

Availability

The pre-trained ELMo-like SeqVec model and a description on how to implement the embeddings into existing methods can be found here: https://github.com/Rostlab/SeqVec . Accessed 10th September 2019.

Predictions on secondary structure, disorder and subcellular localization based on SeqVec can be accessed under: https://embed.protein.properties . Accessed 10th September 2019.

The NetSurfP-2.0 data set [45] used for the evaluation of SeqVec on the task of secondary structure and disorder prediction are publicly available under: http://www.cbs.dtu.dk/services/NetSurfP/ . Accessed 10th September 2019.

The DeepLoc data set [46] used for the evaluation of SeqVec on the task of subcellular localization prediction are publicly available under: http://www.cbs.dtu.dk/services/DeepLoc/data.php . Accessed 10th September 2019.

Funding

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:

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

RO1320/4–1). We gratefully acknowledge the support of NVIDIA Corporation with the donation of two Titan GPU used for this research. We also want to thank the LRZ (Leibniz Rechenzentrum) for providing us access to DGX-V1.

Authors contributions

AE and MH suggested to use ELMo for modeling protein sequences. AE adopted and trained ELMo. MH evaluated SeqVec embeddings on different data sets and tasks. YW helped with discussions about natural language processing. CD implemented the web-interface which allows to access and visualize the predictions. DN helped with various problems regarding the code. FM and BR helped with the design of the experiment and to critically improve the manuscript. MH and AE drafted the manuscript and the other authors provided feedback. All authors read and approved the final manuscript.

Acknowledgements

The authors thank primarily Tim Karl for invaluable help with hardware and software and Inga Weise for support with many other aspects of this work. 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 September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

References

1. Rost B, Sander C: **Prediction of protein secondary structure at better than 70% accuracy**. *Journal of Molecular Biology* 1993, **232**:584-599.
2. Rost B, Sander C: **Improved prediction of protein secondary structure by use of sequence profiles and neural networks**. *Proceedings of the National Academy of Sciences* 1993, **90**:7558-7562.
3. Rost B, Sander C: **Jury returns on structure prediction**. *Nature* 1992, **360**:540.
4. Rost B, Sander C: **Combining evolutionary information and neural networks to predict protein secondary structure**. *Proteins: Structure, Function, and Genetics* 1994, **19**:55-72.
5. Solovyev VV, Salamov AA: **Predicting a-helix and b-strand segments of globular proteins**. *Computer Applications in Biological Science* 1994, **10**:661-669.
6. Barton GJ: **Protein secondary structure prediction**. *Current Opinion in Structural Biology* 1995, **5**:372-376.
7. Chandonia J-M, Karplus M: **Neural networks for secondary structure and structural class predictions**. *Protein Science* 1995, **4**:275-285.
8. Mehta PK, Heringa J, Argos P: **A simple and fast approach to prediction of protein secondary structure from multiply aligned sequences with accuracy above 70%**. *Protein Science* 1995, **4**:2517-2525.
9. Frishman D, Argos P: **Knowledge-based protein secondary structure assignment**. *Proteins: Structure, Function, and Genetics* 1995, **23**:566-579.
10. Jones DT: **Protein secondary structure prediction based on position-specific scoring matrices**. *Journal of Molecular Biology* 1999, **292**(2):195-202.
11. Rost B, Casadio R, Fariselli P, Sander C: **Transmembrane helix prediction at 95% accuracy**. *Protein Science* 1995, **4**:521-533.
12. Rost B, Casadio R, Fariselli P: **Topology prediction for helical transmembrane proteins at 86% accuracy**. *Protein Science* 1996, **5**:1704-1718.
13. Bigelow H, Petrey D, Liu J, Przybylski D, Rost B: **Predicting transmembrane beta-barrels in proteomes**. *Nucleic Acids Research* 2004, **32**:2566-2577.
14. Rost B, Sander C: **Conservation and prediction of solvent accessibility in protein families**. *Proteins: Structure, Function, and Genetics* 1994, **20**(3):216-226.
15. Radivojac P, Obradovic Z, Smith DK, Zhu G, Vucetic S, Brown CJ, Lawson JD, Dunker AK: **Protein flexibility and intrinsic disorder**. *Protein Science* 2004, **13**:71-80.
16. Schlessinger A, Rost B: **Protein flexibility and rigidity predicted from sequence**. *Proteins: Structure, Function, and Bioinformatics* 2005, **61**(1):115-126.
17. Punta M, Rost B: **PROFcon: novel prediction of long-range contacts**. *Bioinformatics* 2005, **21**(13):2960-2968.
18. Peng K, Vucetic S, Radivojac P, Brown CJ, Dunker AK, Obradovic Z: **Optimizing long intrinsic disorder predictors with protein evolutionary information**. *J Bioinform Comput Biol* 2005, **3**(1):35-60.
19. Schlessinger A, Liu J, Rost B: **Natively unstructured loops differ from other loops**. *PLoS Computational Biology* 2007, **3**(7):e140.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

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

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

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

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

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

25. Marino Buslje C, Teppa E, Di Domenico T, Delfino JM, Nielsen M: **Networks of high mutual information define the structural proximity of catalytic sites: implications for catalytic residue identification**. *PLoS Comput Biol* 2010, **6**(11):e1000978.

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

27. Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR: **A method and server for predicting damaging missense mutations**. *Nat Methods* 2010, **7**(4):248-249.

28. Hayat S, Sander C, Marks DS, Elofsson A: **All-atom 3D structure prediction of transmembrane beta-barrel proteins from sequences**. *Proceedings of the National Academy of Sciences of the United States of America* 2015, **112**(17):5413-5418.

29. Marks DS, Hopf TA, Sander C: **Protein structure prediction from sequence variation**. *Nat Biotechnol* 2012, **30**(11):1072-1080.

30. Marks DS, Colwell LJ, Sheridan R, Hopf TA, Pagnani A, Zecchina R, Sander C: **Protein 3D structure computed from evolutionary sequence variation**. *PLoS One* 2011, **6**(12):e28766.

31. Morcos F, Pagnani A, Lunt B, Bertolino A, Marks DS, Sander C, Zecchina R, Onuchic JN, Hwa T, Weigt M: **Direct-coupling analysis of residue coevolution captures native contacts across many protein families**. *Proc Natl Acad Sci U S A* 2011, **108**(49):E1293-1301.

32. Suzek BE, Wang Y, Huang H, McGarvey PB, Wu CH, UniProt C: **UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches**. *Bioinformatics* 2015, **31**(6):926-932.

33. Altschul SF, Madden TL, Schaeffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped Blast and PSI-Blast: a new generation of protein database search programs**. *Nucleic Acids Research* 1997, **25**:3389-3402.

34. Remmert M, Biegert A, Hauser A, Soding J: **HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment**. *Nat Methods* 2012, **9**(2):173-175.

35. Steinegger M, Meier M, Mirdita M, Voehringer H, Haunsberger SJ, Soeding J: **HH-suite3 for fast remote homology detection and deep protein annotation**. *bioRxiv* 2019:560029.

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

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

37. Uversky VN, Radivojac P, Iakoucheva LM, Obradovic Z, Dunker AK: **Prediction of intrinsic disorder and its use in functional proteomics**. *Methods Mol Biol* 2007, **408**:69-92.

38. Dunker AK, Babu MM, Barbar E, Blackledge M, Bondos SE, Dosztanyi Z, Dyson HJ, Forman-Kay J, Fuxreiter M, Gsponer J *et al*: **What's in a name? Why these proteins are intrinsically disordered**. *Intrinsically Disord Proteins* 2013, **1**(1):e24157.

39. Perdigao N, Heinrich J, Stolte C, Sabir KS, Buckley MJ, Tabor B, Signal B, Gloss BS, Hammang CJ, Rost B *et al*: **Unexpected features of the dark proteome**. *Proceedings of the National Academy of Sciences of the United States of America* 2015.

40. Schafferhans A, O'Donoghue SI, Heinzinger M, Rost B: **Dark Proteins Important for Cellular Function**. *Proteomics* 2018, **18**(21-22):e1800227.

41. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L: **Deep contextualized word representations**. *arXiv* 2018, **arXiv:1802.05365**.

42. Schils E, Haan Pd: **Characteristics of sentence length in running text**. *Literary and linguistic computing* 1993, **8**(1):20-26.

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

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

45. Klausen MS, Jespersen MC, Nielsen H, Jensen KK, Jurtz VI, Sonderby CK, Sommer MOA, Winther O, Nielsen M, Petersen B *et al*: **NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning**. *Proteins* 2019.

46. Almagro Armenteros JJ, Sonderby CK, Sonderby SK, Nielsen H, Winther O: **DeepLoc: prediction of protein subcellular localization using deep learning**. *Bioinformatics* 2017, **33**(24):4049.

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

48. Boutet E, Lieberherr D, Tognolli M, Schneider M, Bansal P, Bridge AJ, Poux S, Bougueleret L, Xenarios I: **UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View**. *Methods Mol Biol* 2016, **1374**:23-54.

49. Mikolov T, Chen K, Corrado G, Dean J: **Efficient estimation of word representations in vector space**. *ArXiv* 2013:arXiv:1301.3781.

50. Rives A, Goyal S, Meier J, Guo D, Ott M, Zitnick CL, Ma J, Fergus R: **Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences**. *bioRxiv* 2019:622803.

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

52. Maaten Lvd, Hinton G: **Visualizing data using t-SNE**. *Journal of machine learning research* 2008, **9**(Nov):2579-2605.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

53. Fox NK, Brenner SE, Chandonia J-M: **SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures**. *Nucleic acids research* 2013, **42**(D1):D304-D309.

54. Kosloff M, Kolodny R: **Sequence-similar, structure-dissimilar protein pairs in the PDB**. *Proteins: Structure, Function, and Bioinformatics* 2008, **71**(2):891-902.

55. Bairoch A: **The ENZYME database in 2000**. *Nucleic acids research* 2000, **28**(1):304-305.

56. Velankar S, Dana JM, Jacobsen J, Van Ginkel G, Gane PJ, Luo J, Oldfield TJ, O'donovan C, Martin M-J, Kleywegt GJ: **SIFTS: structure integration with function, taxonomy and sequences resource**. *Nucleic acids research* 2012, **41**(D1):D483-D489.

57. Heffernan R, Yang Y, Paliwal K, Zhou Y: **Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility**. *Bioinformatics* 2017, **33**(18):2842-2849.

58. Wang S, Peng J, Ma J, Xu J: **Protein secondary structure prediction using deep convolutional neural fields**. *Scientific reports* 2016, **6**:18962.

59. Wang S, Li W, Liu S, Xu J: **RaptorX-Property: a web server for protein structure property prediction**. *Nucleic acids research* 2016, **44**(W1):W430-W435.

60. Drozdetskiy A, Cole C, Procter J, Barton GJ: **JPred4: a protein secondary structure prediction server**. *Nucleic acids research* 2015, **43**(W1):W389-W394.

61. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: **The protein data bank**. *Nucleic acids research* 2000, **28**(1):235-242.

62. Wang G, Dunbrack Jr RL: **PISCES: a protein sequence culling server**. *Bioinformatics* 2003, **19**(12):1589-1591.

63. Yang Y, Gao J, Wang J, Heffernan R, Hanson J, Paliwal K, Zhou Y: **Sixty-five years of the long march in protein secondary structure prediction: the final stretch?** *Briefings in bioinformatics* 2016, **19**(3):482-494.

64. Cuff JA, Barton GJ: **Evaluation and improvement of multiple sequence methods for protein secondary structure prediction**. *Proteins: Structure, Function, and Genetics* 1999, **34**(4):508-519.

65. Abriata LA, Tamò GE, Monastyrskyy B, Kryshtafovych A, Dal Peraro M: **Assessment of hard target modeling in CASP12 reveals an emerging role of alignment-based contact prediction methods**. *Proteins: Structure, Function, and Bioinformatics* 2018, **86**:97-112.

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

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

68. Briesemeister S, Blum T, Brady S, Lam Y, Kohlbacher O, Shatkay H: **SherLoc2: a high-accuracy hybrid method for predicting subcellular localization of proteins**. *J Proteome Res* 2009, **8**(11):5363-5366.

69. Yu CS, Chen YC, Lu CH, Hwang JK: **Prediction of protein subcellular localization**. *Proteins* 2006, **64**(3):643-651.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

70. Horton P, Park KJ, Obayashi T, Fujita N, Harada H, Adams-Collier CJ, Nakai K: **WoLF PSORT: protein localization predictor**. *Nucleic Acids Res* 2007, **35**(Web Server issue):W585-587.

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

72. Li W, Godzik A: **Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences**. *Bioinformatics* 2006, **22**(13):1658-1659.

73. Fu L, Niu B, Zhu Z, Wu S, Li W: **CD-HIT: accelerated for clustering the next-generation sequencing data**. *Bioinformatics* 2012, **28**(23):3150-3152.

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

75. Bailey TL, Boden M, Buske FA, Frith M, Grant CE, Clementi L, Ren J, Li WW, Noble WS: **MEME SUITE: tools for motif discovery and searching**. *Nucleic Acids Res* 2009, **37**(Web Server issue):W202-208.

76. Bernard G, Chan CX, Ragan MA: **Alignment-free microbial phylogenomics under scenarios of sequence divergence, genome rearrangement and lateral genetic transfer**. *Sci Rep* 2016, **6**:28970.

77. Nakai K, Horton P: **PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization**. *Trends Biochem Sci* 1999, **24**(1):34-36.

78. Kuang R, Ie E, Wang K, Wang K, Siddiqi M, Freund Y, Leslie C: **Profile-based string kernels for remote homology detection and motif extraction**. *J Bioinform Comput Biol* 2005, **3**(3):527-550.

79. Leslie C, Eskin E, Weston J, Noble WS: **Mismatch string kernels for SVM protein classification**. *Bioinformatics* 2003:in press.

80. Noble WS, Kuang R, Leslie C, Weston J: **Identifying remote protein homologs by network propagation**. *FEBS J* 2005, **272**(20):5119-5128.

81. Hamp T, Rost B: **Evolutionary profiles improve protein-protein interaction prediction from sequence**. *Bioinformatics* 2015, **31**(12):1945-1950.

82. Asgari E, McHardy AC, Mofrad MRK: **Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX)**. *Sci Rep* 2019, **9**(1):3577.

83. Kim S, Lee H, Kim K, Kang J: **Mut2Vec: distributed representation of cancerous mutations**. *BMC medical genomics* 2018, **11**(2):33.

84. Xu Y, Song J, Wilson C, Whisstock JC: **PhosContext2vec: a distributed representation of residue-level sequence contexts and its application to general and kinase-specific phosphorylation site prediction**. *Scientific reports* 2018, **8**.

85. Bojanowski P, Grave E, Joulin A, Mikolov T: **Enriching word vectors with subword information**. *Transactions of the Association for Computational Linguistics* 2017, **5**:135-146.

86. Pennington J, Socher R, Manning C: **Glove: Global vectors for word representation**. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP): 2014*. 1532-1543.

bioRxiv preprint doi: https://doi.org/10.1101/614313; this version posted September 10, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

87. Devlin J, Chang M-W, Lee K, Toutanova K: **Bert: Pre-training of deep bidirectional transformers for language understanding**. *arXiv preprint arXiv:181004805* 2018.

88. Kim Y, Jernite Y, Sontag D, Rush AM: **Character-aware neural language models**. In: *Thirtieth AAAI Conference on Artificial Intelligence: 2016*.

89. Reddi SJ, Kale S, Kumar S: **On the convergence of adam and beyond**. *arXiv preprint arXiv:190409237* 2019.

90. Kingma DP, Ba J: **Adam: A method for stochastic optimization**. *arXiv preprint arXiv:14126980* 2014.

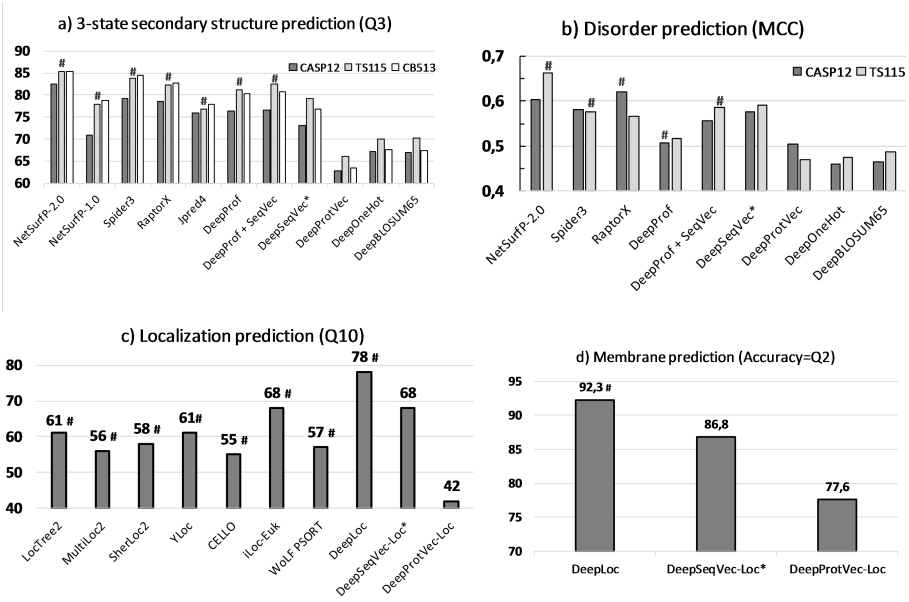
91. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R: **Dropout: a simple way to prevent neural networks from overfitting**. *The Journal of Machine Learning Research* 2014, **15**(1):1929-1958.

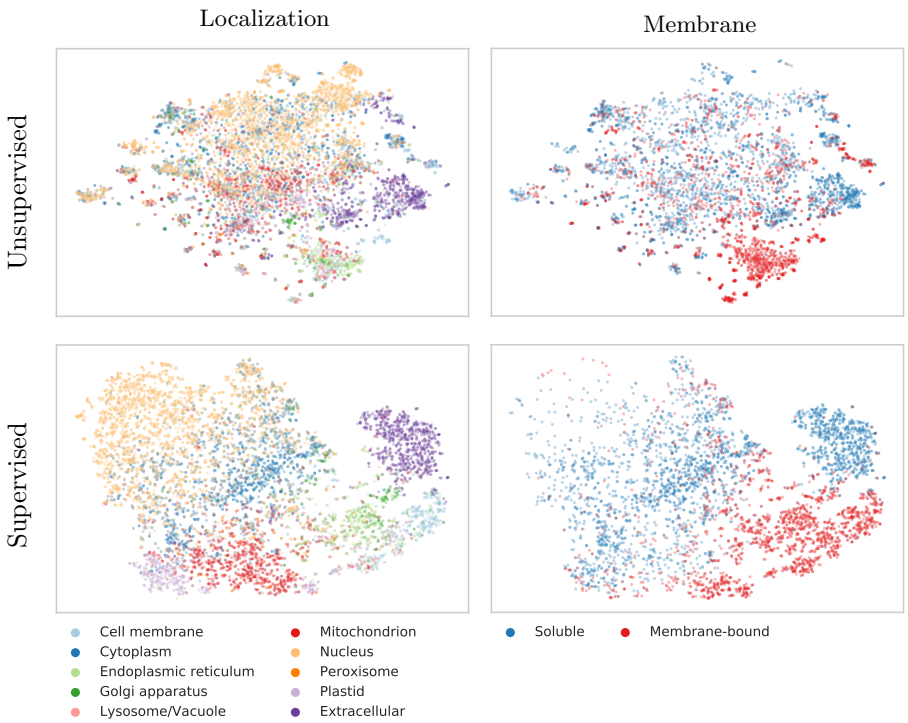
92. Henikoff S, Henikoff JG: **Amino acid substitution matrices from protein blocks**. *PNAS* 1992, **89**:10915-10919.

93. Ioffe S, Szegedy C: **Batch normalization: Accelerating deep network training by reducing internal covariate shift**. *arXiv preprint arXiv:150203167* 2015.

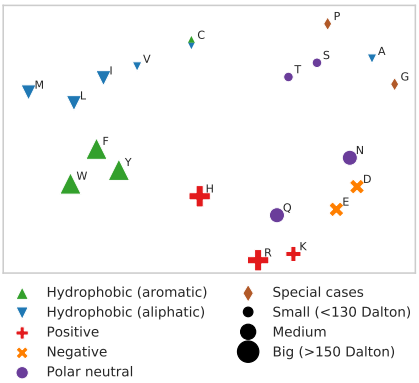
94. Matthews BW: **Comparison of the predicted and observed secondary structure of T4 phage lysozyme**. *Biochimica et Biophysica Acta* 1975, **405**:442-451.

95. Gorodkin J: **Comparing two K-category assignments by a K-category correlation coefficient**. *Computational biology and chemistry* 2004, **28**(5-6):367-374.

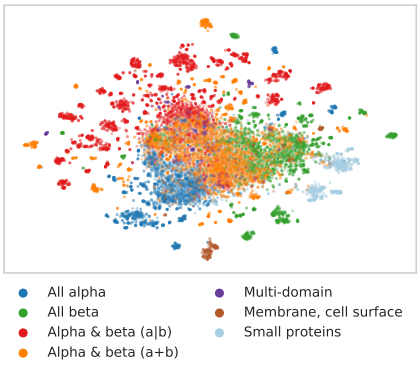

a) 3-state secondary structure prediction (Q3)
CASP12
TS115
CB513
NetSurfP-2.0
NetSurfP-1.0
Spider3
RaptorX
Jpred4
DeepProf
DeepProf + SeqVec
DeepSeqVec*
DeepProtVec
DeepOneHot
DeepBLOSUM65
b) Disorder prediction (MCC)
CASP12
TS115
c) Localization prediction (Q10)
61 #
56 #
58 #
61#
55 #
68 #
57 #
78 #
68
42
LocTree2
MultiLoc2
SherLoc2
YLoc
CELLO
iLoc-Euk
WoLF PSORT
DeepLoc
DeepSeqVec-Loc*
DeepProtVec-Loc
d) Membrane prediction (Accuracy=Q2)
92,3 #
86,8
77,6
DeepLoc
DeepSeqVec-Loc*
DeepProtVec-Loc


Localization
Membrane
Unsupervised
Supervised
Cell membrane
Cytoplasm
Endoplasmic reticulum
Golgi apparatus
Lysosome/Vacuole
Mitochondrion
Nucleus
Peroxisome
Plastid
Extracellular
Soluble
Membrane-bound

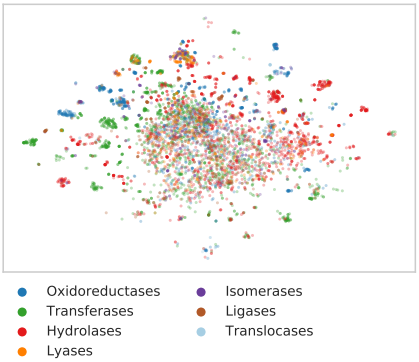
(a) Amino acids



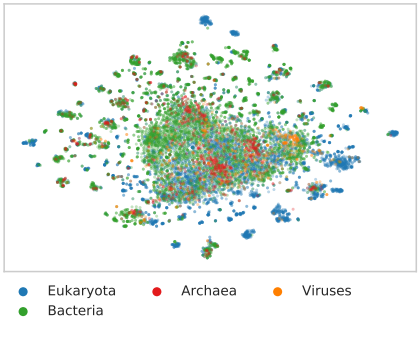
(b) SCOPe

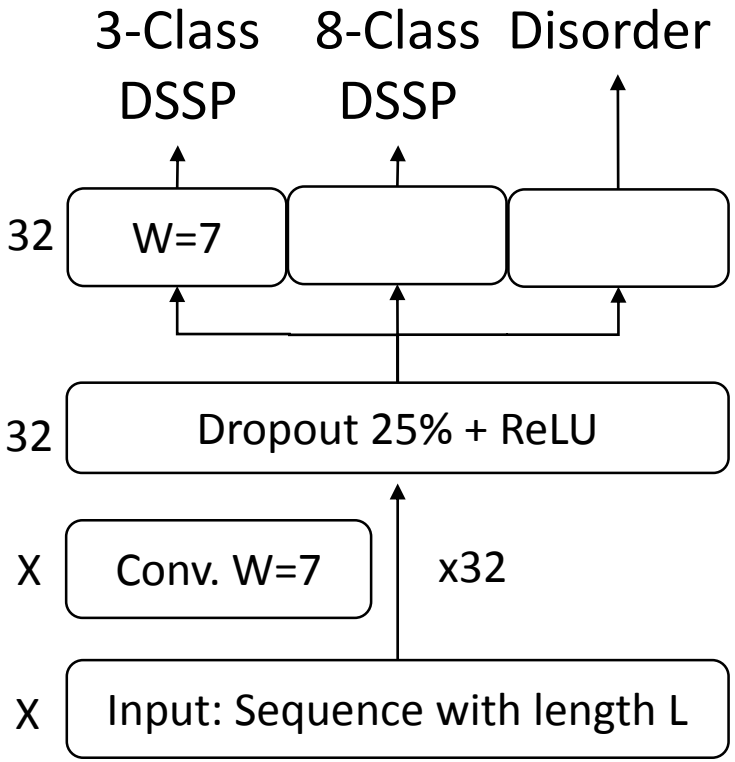


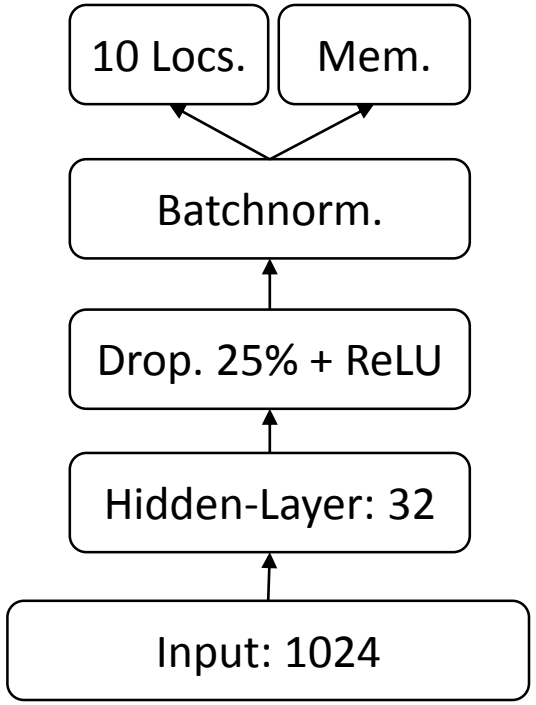
(c) Enzyme Commission number (E.C.)



(d) Kingdoms




3-Class DSSP
8-Class DSSP
Disorder
32
W=7
32
Dropout 25% + ReLU
X
Conv. W=7
x32
X
Input: Sequence with length L


10 Locs.
Mem.
Batchnorm.
Drop. 25% + ReLU
Hidden-Layer: 32
Input: 1024