Light Attention Predicts Protein Location from the Language of Life

Hannes Stärk [*1] **Christian Dallago** [*1 2] **Michael Heinzinger** [1 2] **Burkhard Rost** [1 3 4]

Abstract

Although knowing where a protein functions in a cell is important to characterize biological processes, this information remains unavailable for most known proteins. Machine learning narrows the gap through predictions from expertly chosen input features leveraging evolutionary information that is resource expensive to generate. We showcase using embeddings from protein language models for competitive localization predictions not relying on evolutionary information. Our lightweight deep neural network architecture uses a softmax weighted aggregation mechanism with linear complexity in sequence length referred to as light attention (LA). The method significantly outperformed the state-of-the-art for ten localization classes by about eight percentage points (Q10). The novel models are available as a web-service and as a stand-alone application at embed.protein.properties.

1. Introduction

Proteins are the machinery of life involved in all essential biological processes (biological background in Appendix). Knowing where in the cell a protein functions, referred to as its *subcellular localization* or *cellular compartment*, is important for unraveling biological function (Nair & Rost, 2005; Yu et al., 2006). Experimental determination of protein function is complex, costly, and selection biased (Ching et al., 2018). In contrast, the costs of determining protein sequences continuously decrease (Consortium, 2021), increasing the sequence-annotation gap (gap between proteins of known sequence and unknown function). Computational methods have been bridging this gap (Rost et al., 2003); one way has been to predict protein subcellular location (Goldberg et al., 2012; 2014; Almagro Armenteros et al., 2017; Savojardo et al., 2018). The standard tool in molecular biology, namely homology-based inference (HBI), accurately transfers annotations from experimentally annotated to sequence-similar un-annotated proteins. However, HBI is not available or unreliable for most proteins (Goldberg et al., 2014; Mahlich et al., 2018). Machine learning methods perform less well (lower precision) but are available for all proteins (high recall). The best methods use evolutionary information from families of related proteins as input (Goldberg et al., 2012; Almagro Armenteros et al., 2017). Although the marriage of evolutionary information and machine learning has influenced computational biology for decades (Rost & Sander, 1993), due to database growth, this information becomes increasingly costly to generate.

Recently, protein sequence representations (embeddings) have been learned from databases (Steinegger & Söding, 2018; Consortium, 2021) using language models (LMs) (Heinzinger et al., 2019; Rives et al., 2019; Alley et al., 2019; Elnaggar et al., 2020) initially used in natural language processing (NLP) (Radford, 2018; Devlin et al., 2019; Radford et al., 2019). Models trained on protein embeddings via transfer learning tend to be outperformed by approaches using evolutionary information (Rao et al., 2019; Heinzinger et al., 2019). However, embedding-based solutions can even outshine HBI (Littmann et al., 2021) and models predicting aspects of protein structure (Bhattacharya et al., 2020; Rao et al., 2020). Yet, for location prediction, embedding-based models (Heinzinger et al., 2019; Elnaggar et al., 2020; Littmann et al., 2021) remained inferior to the state-of-the-art using evolutionary information, e.g., represented by DeepLoc (Almagro Armenteros et al., 2017).

In this work, we leveraged protein embeddings to predict cellular location without evolutionary information. We proposed a deep neural network architecture using light attention (LA) inspired by previous attention mechanisms (Bahdanau et al., 2015; Vaswani et al., 2017).

[*]Equal contribution [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/Munich, Germany [3]Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany [4]TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany. Correspondence to: Christian Dallago <christian.dallago@tum.de>.

2. Related Work

Previous state-of-the-art (SOTA) models for subcellular location prediction combined homology, evolutionary information, and machine learning, often building prior knowledge about biology into model architectures. For instance, LocTree2 (Goldberg et al., 2012) implemented profile-kernel SVMs (Cortes & Vapnik, 1995; Rui Kuang et al., 2004) which identified k-mers conserved in evolution and put them into a hierarchy of models inspired by cellular sorting pathways. BUSCA (Savojardo et al., 2018) combines three compartment-specific prediction methods based on SVMs using evolutionary information (Pierleoni et al., 2006; 2011; Savojardo et al., 2017). DeepLoc (Almagro Armenteros et al., 2017) uses convolutions followed by a bidirectional LSTM (Hochreiter & Schmidhuber, 1997; Schuster & Paliwal, 1997) that employs Bahdanau-Attention (Bahdanau et al., 2015). Using evolutionary information either in the form of residue substitution scores (Henikoff & Henikoff, 1992) or protein sequence profiles (Gribskov et al., 1987), DeepLoc rose to become the SOTA. Embedding-based methods (Heinzinger et al., 2019) have not yet outperformed this SOTA, although ProtTrans (Elnaggar et al., 2020), based on very large data sets, came close.

3. Methods

3.1. Data

Standard set DeepLoc. Following previous work (Heinzinger et al., 2019; Elnaggar et al., 2020), we mainly used a data set introduced by *DeepLoc* (Almagro Armenteros et al., 2017) for training and testing. The training set contained 13 858 proteins annotated with experimental evidence for one of ten location classes (nucleus, cytoplasm, extracellular space, mitochondrion, cell membrane, Endoplasmatic Reticulum, plastid, Golgi apparatus, lysosome/vacuole, peroxisome). Another 2 768 proteins made up the test set (henceforth called *setDeepLoc*), which had been redundancy reduced to the training set (but not to itself) at 30% pairwise sequence identity (PIDE) or to an E-value cutoff of 10^{-6}. To tune model parameters and avoid overestimating performance, we further split the DeepLoc training set into a training set containing 9 503 sequences and a validation set (redundancy reduced to training by 30% PIDE) containing 1 158 sequences (distribution of classes in *Appendix: Datasets*).

Novel *setHARD*. To rule out that methods had been optimized for the static standard test set (*setDeepLoc*) used by many developers, we created a new independent test set from SwissProt (Consortium, 2021). Applying the same filtering mechanisms as the DeepLoc developers (only eukaryotes; only proteins longer than 40 residues; no fragments; only experimental location annotations) gave 5 947 proteins. Using MMseqs2 (Steinegger & Söding, 2017), we removed all proteins from the new set with more than 20% PIDE to any protein in DeepLoc (both training and testing data). Next, we mapped location classes from DeepLoc to SwissProt, merged duplicates, and removed multi-localized proteins (protein X both in class Y and Z). Finally, we clustered proteins to representatives at 20% PIDE and obtained a new and more challenging test set (dubbed *setHARD*) with 490 proteins. Class distributions differed between the two sets (see *Appendix: Datasets*).

Table 1. Parameters and implementation details of SeqVec (Heinzinger et al., 2019), ProtBert and ProtT5 (Elnaggar et al., 2020). The time it takes to embed a single sequence (sec per sequence) is averaged over embedding 10 000 proteins taken from the Protein Data Bank (PDB) (Berman et al., 2000). The number of sequences used for the pre-training task is detailed in "# sequences".

	SeqVec	ProtBert	ProtT5
parameters	93M	420M	3B
# sequences	33M	2.1B	2.1B
Sec per sequence	0.03	0.06	0.1
attention heads	-	16	32
float precision	32bit	32bit	16bit
size (GB)	0.35	1.6	2.75

3.2. Models

Input: protein embeddings. As input to the LA architectures, we extracted embeddings from three protein language models (LMs; Table 1): the bidirectional LSTM *SeqVec* (Heinzinger et al., 2019) based on ELMo (Peters et al., 2018) trained on UniRef50 (Suzek et al., 2015), the encoder-only model *ProtBert* (Elnaggar et al., 2020) based on BERT (Devlin et al., 2019) trained on BFD (Steinegger & Söding, 2018), and the encoder-only model ProtT5-XL-UniRef50 (Elnaggar et al., 2020) (for simplicity: *ProtT5*) based on T5 (Raffel et al., 2020) trained on BFD and fine-tuned on Uniref50. *ProtT5* was instantiated at half-precision (float16 weights instead of float32) to ensure the encoder could fit on consumer GPUs with limited vRAM. Embeddings for each residue (NLP equivalent: word) in a protein sequence (NLP equivalent: document) were obtained using the bio-embeddings software (Dallago et al., 2021). For *SeqVec*, the per-residue embeddings were generated by summing the representations of each layer. For *ProtBert* and *ProtT5*, the per-residue embeddings were extracted from the last hidden layer of the models. Finally, the inputs obtained from the protein LMs were of size $1024 \times L$, where L is the length of the protein sequence.

Light Attention (LA) architecture. The input to the light attention (LA) classifier (Figure 1) was a protein embedding

$x \in \mathbb{R}^{1024 \times L}$ where L is the sequence length. The input was transformed by two separate 1D convolutions with filter sizes s and learned weights $W^{(e)}, W^{(v)} \in \mathbb{R}^{s \times 1024 \times d_{out}}$. The convolutions were applied over the length dimension to produce attention coefficients and values $e, v \in \mathbb{R}^{d_{out} \times L}$

$$e_{i,j} = b_i + \sum_{k=1}^{1024} \sum_{l=-\lfloor \frac{s}{2} \rfloor}^{\lceil \frac{s}{2} \rceil} W_{l,i}^{(e)} x_{:,j+l} \tag{1}$$

where $b \in \mathbb{R}^{d_{out}}$ is a learned bias and $x_{:,j}$ denotes the j-th residue embedding. For $j \notin [0, L)$, the $x_{:,j}$ were zero vectors. To use the coefficients as attention distributions over all j, we softmax-normalized them over the length dimension. The attention weight $\alpha_{i,j} \in \mathbb{R}$ for the j-th residue and the i-th feature dimension was calculated as:

$$\alpha_{i,j} = \frac{exp(e_{i,j})}{\sum_{l=1}^{L} exp(e_{i,l})} \tag{2}$$

Note that the weight distributions for each feature dimension i are independent, and they can generate different attention patterns. The attention distributions were used to compute weighted sums of the transformed residue embeddings $v_{i,j}$. Thus, we obtained a fixed-size representation $x' \in \mathbb{R}^{d_{out}}$ for the whole protein, independent of its length.

$$x'_i = \sum_{j=1}^{L} \alpha_{i,j} v_{i,j} \tag{3}$$

We concatenated x'_i with the maximum of the values over the length dimension $v^{max} \in \mathbb{R}^{d_{out}}$, meaning $v_i^{max} = \max_{1 \leq j \leq L}(v_{i,j})$. This concatenated vector was input into a two layer multi-layer perceptron (MLP) $f : \mathbb{R}^{2d_{out}} \mapsto \mathbb{R}^{d_{class}}$ with d_{class} as the number of classes. The softmax over the MLP output represents the class probabilities indexed by c ($\oplus$ denotes concatenation):

$$p(c|x) = softmax_c(f(x' \oplus m)) \tag{4}$$

Implementation details. The LA models were trained using filter size $s = 9$, $d_{out} = 1024$, the Adam (Kingma & Ba, 2015) optimizer (learning rate 5×10^{-5}) with a batch size of 150, and early stopping after no improvement in validation loss for 80 epochs. We selected the hyperparameters via random search (*Appendix: Hyperparameters*). Training was done on either an Nvidia Quadro RTX 8000 with 48GB vRAM or an Nvidia GeForce GTX 1060 with 6GB vRAM.

Methods used for comparison. For comparison, we trained a two-layer MLP proposed previously (Heinzinger et al., 2019). Instead of per-residue embeddings in $\mathbb{R}^{1024 \times L}$, the MLPs used sequence-embeddings in $\mathbb{R}^{1024}$, which derived from residue-embeddings averaged over the length dimension (i.e. mean pooling). Furthermore, for these representations, we performed annotation transfer (dubbed AT) based on embedding similarity (Littmann et al., 2021). In this approach, proteins in *setDeepLoc* and *setHARD* were annotated by transferring the class of the nearest neighbor in the DeepLoc training set (given by L1 distance).

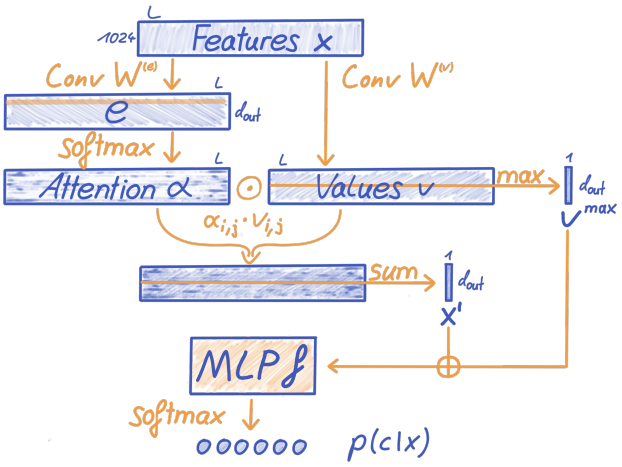

Figure 1. **Sketch of LA solution.** The LA architecture is parameterized by two weight matrices $W^{(e)}, W^{(v)} \in \mathbb{R}^{s \times 1024 \times d_{out}}$ and the weights of an MLP $f : \mathbb{R}^{2d_{out}} \mapsto \mathbb{R}^{d_{class}}$.

3.3. Evaluation.

Following previous work, we assessed performance through the mean ten-class accuracy (Q10), giving the percentage of correctly predicted proteins in one of ten location classes. Additional measures tested (e.g., F1 score and Matthew's correlation coefficient (MCC) for multiple classes (Gorodkin, 2004)) did not provide any additional insights and were confined to the *Appendix: Additional Results*. Error estimates were calculated over ten random seeds on both test sets. For previous methods (DeepLoc and DeepLoc62 (Almagro Armenteros et al., 2017), LocTree2 (Goldberg et al., 2012), MultiLoc2 (Blum et al., 2009), SherLoc2 (Briesemeister et al., 2009), CELLO (Yu et al., 2006), iLoc-Euk (Chou et al., 2011), YLoc (Briesemeister et al., 2010) and WoLF PSORT (Horton et al., 2007)) published performance values were used (Almagro Armenteros et al., 2017) for *setDeepLoc*. For *setHARD*, the webserver for DeepLoc[1] was used to generate predictions using either profile or BLOSUM inputs, whose results were later evaluated in Q10 and MCC. The majority classifier was used as a naive baseline. All evaluation scripts to reproduce results are available[2].

[1] http://www.cbs.dtu.dk/services/DeepLoc

[2] https://github.com/HannesStark/protein-localization

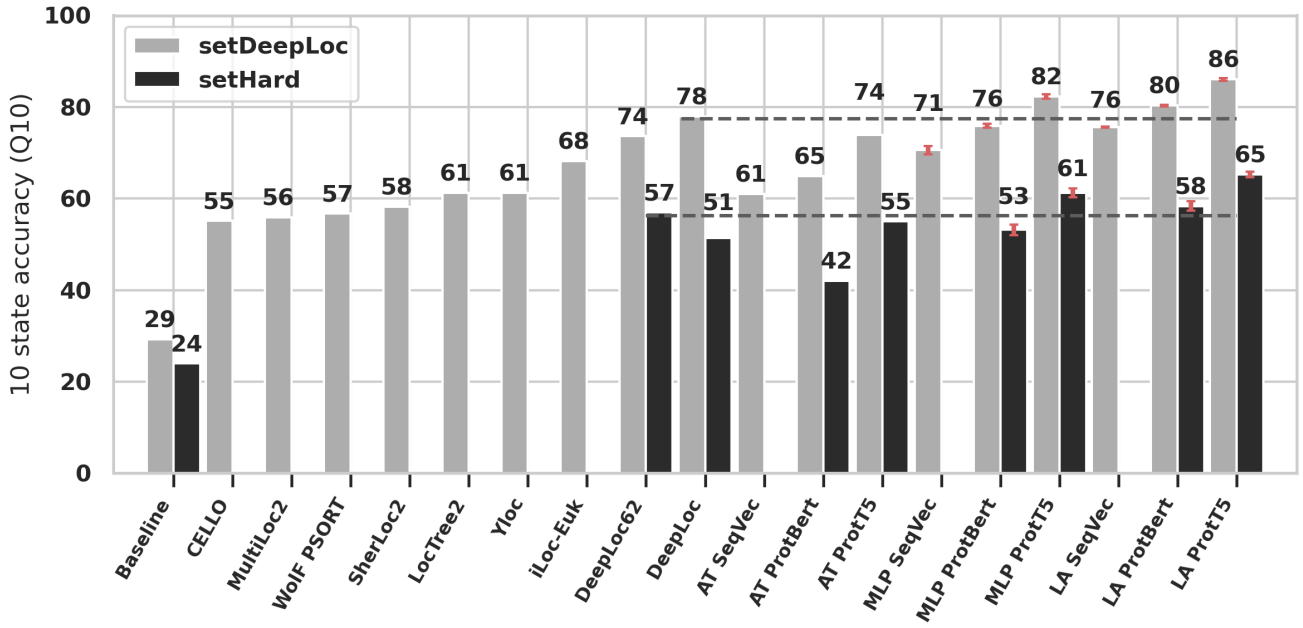

Figure 2. **LA architectures perform best.** Bars give the ten-class accuracy (Q10) for popular location prediction methods on *setDeepLoc* (light-gray bars) and *setHARD* (dark-gray bars). Baseline is the most common class in each set. Horizontal gray dashed lines mark the previous SOTA on either set. Estimates for standard errors are marked in orange for the methods introduced here. *setHARD* results are provided for a subset of methods that yielded the best results on *setDeepLoc* (see *Methods* for detail on the external methods used; tabular data in *Appendix: Additional Results*). Two results stood out: (i) the LA approaches introduced here outperformed the top methods although not using evolutionary information (highest bars), and (ii) the performance estimates differed completely between the two data sets (difference light/dark gray).

4. Results

Embeddings outperformed evolutionary information. The simple AT approach already outperformed some methods using evolutionary information in the form of sequence profiles or BLOSUM62 encoding (Henikoff & Henikoff, 1992) (Figure 2: *AT**). The MLP trained on *ProtT5* (Elnaggar et al., 2020) outperformed *DeepLoc* (Almagro Armenteros et al., 2017) (Figure 2: *MLP ProtT5* vs. *DeepLoc*). Methods based on *ProtT5* embeddings consistently yielded better results than *ProtBert* and *SeqVec* (Heinzinger et al., 2019) (**ProtT5* vs **ProtBert/*SeqVec* in Figure 2). Results on Q10 are consistent with MCC (*Appendix: Additional Results*).

LA architecture best. The light attention (LA) architecture consistently outperformed other embedding-based approaches, irrespective of the protein LM used (LA* vs. AT/MLP* in Figure 2). Using *ProtBert* embeddings, LA outperformed the state-of-the-art (Almagro Armenteros et al., 2017) by 1 and 2 percentage points on *setHARD* and *setDeepLoc* (*LA ProtBert* Figure 2). For both test sets, LA improved the previous best on either set by around 8 percentage points when using *ProtT5* embeddings.

Overfitting by using standard data set. The substantial drop in performance (around 22 percentage points) between results for the standard *setDeepLoc* and the new challenging *setHARD* (Figure 2: light-gray vs. dark-gray, respectively) suggests some level of overfitting. Mimicking the distribution of classes found in *setDeepLoc* by sampling with replacement from *setHARD* led to better results (in Q10: *DeepLoc62*=63%; *DeepLoc*=54%; *LA ProtBert*=62%; *LA ProtT5*=69%). *DeepLoc* performed worse on *setHARD* using profiles than when using simple sequence information/BLOSUM (Figure 2: *DeepLoc* vs. *DeepLoc62*). Otherwise, the relative ranking and difference of models largely remained consistent between *setDeepLoc* and *setHARD*.

Low performance for minority classes. The confusion matrix of predictions for *setDeepLoc* using LA trained on *ProtT5* embeddings highlighted how many proteins were incorrectly predicted in the most prevalent class, *cytoplasm*, and that even the two majority classes were often confused with each other (Figure 3: *nucleus* and *cytoplasm*). In line with the previous SOTA (Almagro Armenteros et al., 2017), the performance was particularly low for the most under-represented classes, namely *Golgi apparatus* , *lysosome/Vacuole*, and *peroxisome* (accounting for 2.6%, 2.3%, and 1.1% of the data, respectively).

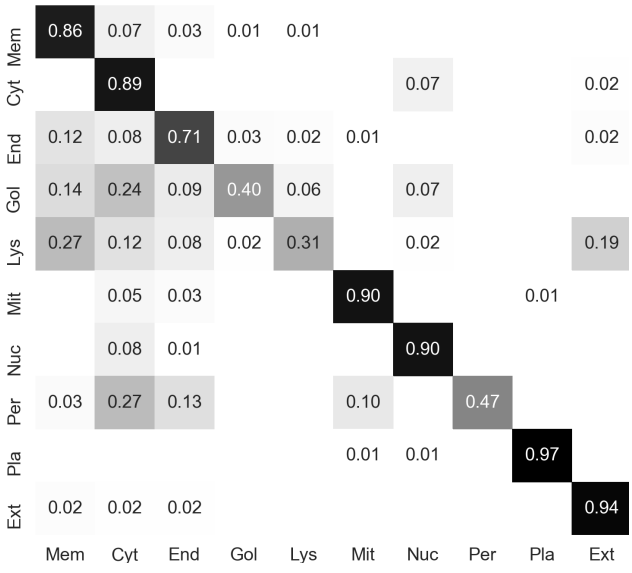

Figure 3. **Mostly capturing majority classes.** Confusion matrix of LA predictions on ProtT5 (Elnaggar et al., 2020) embeddings for *setDeepLoc* (Almagro Armenteros et al., 2017) (*setHARD* in Appendix) annotated with the fraction of the true class. Vertical axis: true class, horizontal axis: predicted class. Labels: Mem=cell **Mem**brane; Cyt=**Cyt**oplasm; End=**End**oplasmatic Reticulum; Gol=**Gol**gi apparatus; Lys=**Lys**osome/vacuole; Mit=**Mit**ochondrion; Nuc=**Nuc**leus; Per=**Per**oxisome; Pla=**Pla**stid; Ext=**Ext**racellular

Table 2. **LA + ProtT5 the winning combination.** Accuracy of baselines and ablations using *ProtT5* embeddings (above the line), one-hot residue encodings or profiles for *setDeepLoc* and *setHARD* for various architectures. LA ProtT5: The proposed light attention architecture. LA - Softmax: replaced softmax aggregation that previously produced x' with averaging of the coefficients e over the length dimension. LA - MaxPool: discarded max-pooled values v^{max} as input to the MLP, aka. only the softmax aggregated features x' were used. Attention from v: attention coefficients e were obtained via a convolution over the values v instead of over the inputs x. DeepLoc LSTM: the architecture of DeepLoc (Almagro Armenteros et al., 2017) was used instead of LA. Conv + AdaPool: a stack of convolutions (kernel-size 3, 9, and 15) followed by adaptive pooling to a length of 5 and an MLP was used instead of LA. LA on OneHot: LA using one-hot encodings of residues in a protein sequence as input. LA on Profiles: LA using evolutionary information in the form of protein profiles (Gribskov et al., 1987) as input.

METHOD	SETDEEPLOC	SETHARD
LA PROTT5	**86.01**± 0.34	**65.21**± 0.61
LA - SOFTMAX	85.30± 0.32	64.72± 0.70
LA - MAXPOOL	84.79± 0.19	63.84± 0.67
ATTENTION FROM v	85.41± 0.27	64.77± 0.93
DEEPLOC LSTM	79.40± 0.88	59.36± 0.84
CONV + ADAPOOL	82.09± 0.92	60.79± 2.01
LA ON ONEHOT	43.53± 1.48	32.57± 2.38
LA ON PROFILES	43.78± 1.25	33.35± 1.82

Light aggregation (LA) mechanism crucial. To probe the effectiveness of LA's aggregation mechanism on *ProtT5* embeddings, a number of tests were run to consider both architecture changes (Table 2: *LA - Softmax* & *LA - MaxPool* & *Attention from v* & *DeepLoc LSTM* & *Conv + AdaPool*) and input changes (Table 2: *LA on OneHot* & *LA on Profiles*). Performance deterioration by dropping the softmax (Table 2: *LA - Softmax*) or max-pooling aggregation (Table 2: *LA - MaxPool*) confirmed that both aspects are crucial and lead to better performance. Furthermore, LA is especially apt at extracting information from LM embeddings, while it performs poorly on other protein representations, e.g., one-hot encodings (Table 2: *LA on OneHot*) or profiles (Table 2: *LA on Profiles*).

Model trainable on consumer hardware. After embeddings for proteins were generated, the final LA architecture, made of 18 940 224 parameters, could be trained on an Nvidia GeForce GTX 1060 with 6GB vRAM in 18 hours or on a Quadro RTX 8000 with 48GB vRAM in 2.5 hours.

5. Discussion

Light attention beats pooling. The central challenge for the improvement introduced here was to convert the residue-embeddings (NLP equivalent: word embeddings) from protein language models such as *SeqVec* (Heinzinger et al., 2019), *ProtBert*, or *ProtT5* (Elnaggar et al., 2020) to meaningful sequence-embeddings (NLP equivalent: document). A qualitative evaluation of the influence of the attention mechanism (Figure 4) highlighted its ability to steer predictions. Although averaging surpassed evolutionary-information-based methods using simple similarity-based annotation transfer (Figure 2: AT*) and in one instance even SOTA using a simple feed-forward network (Figure 2: *DeepLoc* vs. *MLP ProtT5*), LA was able to consistently distill more information from embeddings. Most likely, the improvement can be attributed to LA's ability to regulate the immense difference in lengths of proteins (varying from 30 to 30 000 residues (Consortium, 2021)) by learning attention distributions over the sequence positions. LA models appeared to have captured relevant long-range dependencies while retaining the ability to focus on specific sequence regions such as beginning and end, which play a particularly important role in determining protein location for some proteins (Lange et al., 2007; Almagro Armenteros et al., 2017).

First win over evolutionary information. Effectively, LA trained on protein LM embeddings from *ProtT5* (Elnaggar et al., 2020) was at the heart of the first method that clearly appeared to outperform the best existing method (*DeepLoc*,

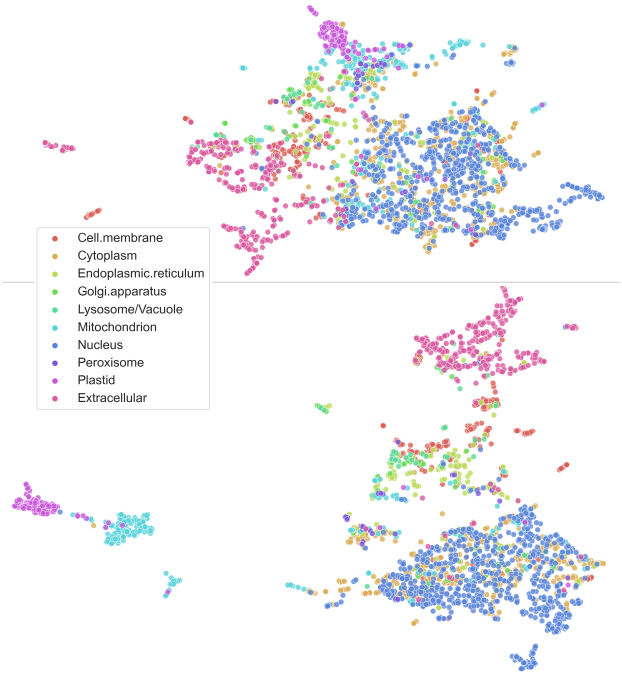

Figure 4. **Qualitative analysis confirms: attention effective.** UMAP (McInnes et al., 2018) projections of per-protein embeddings colored according to subcellular location (*setDeepLoc*). Both plots were created with the same default values of the python *umap-learn* library. Top: ProtT5 embeddings (LA input; x) mean-pooled over protein length (as for MLP/AT input). Bottom: ProtT5 embeddings (LA input; x) weighted according to the attention distribution produced by LA (this is not x' as we sum the input features x and not the values v after the convolution).

(Almagro Armenteros et al., 2017; Heinzinger et al., 2019)) in a statistically significant manner on two test sets (Figure 2). To the best of our knowledge, this improvement was the first instance ever that embedding-based transfer learning substantially outperformed AI/ML methods using evolutionary information (in the form of protein profiles or BLOSUM62 encoding in *DeepLoc's* case) for function prediction. Even if embeddings are extracted from LMs trained on large sequence data originating from evolution, the vast majority of data learned originates from much more generic constraints informative of protein structure and function.

Better and faster than profiles. At inference, the embeddings needed as input for the LA models come with three advantages over the historically most informative evolutionary information, i.e., protein profiles, which were essential for methods such as *DeepLoc* (Almagro Armenteros et al., 2017) to achieve SOTA. Chiefly, embeddings can be obtained in far less time than is needed to generate profiles and require fewer compute resources. Even the lightning-fast MMseqs2 (Steinegger & Söding, 2017), which is not the standard in bioinformatics (other methods 10-100x slower), in our experience, required about 0.3 seconds per sequence to generate profiles for a large set of 10 000 proteins. The slowest but most informative protein LM (*ProtT5*) is 3x faster, while the second most informative (*ProtBert*) is 5x faster (Table 1). Moreover, these MMseqs2 stats derive from runs on a machine with $>$ 300GB of RAM and 2x40cores/80threads CPUs, while generating LM embeddings required only a moderate machine (8 cores, 16GB RAM) equipped with a modern GPU with $>$7GB of vRAM. Additionally, extracting profiles relies on the use of tools (e.g., MMseqs2) that are sensitive to parameter changes, ultimately an extra complication for users. In contrast, generating embeddings doesn't require a parameter choice beyond which trained model to use (e.g., *ProtBert* vs. *ProtT5*). However, retrieving less informative evolutionary information (BLOSUM (Urban et al., 2020)) consists of a simple hash-table lookup. As such, up to implementation and optimizations, computing this type of information is instantaneous, beating even the fastest protein LM *SeqVec*. One downside to using embeddings is one-off expensive language model pre-training (Elnaggar et al., 2020; Heinzinger et al., 2019). Yet, compared to many-times expensive profile generation, the cost may be justified and absorbed.

Overfitting through standard data set? For protein subcellular location prediction, the data set of *DeepLoc* (Almagro Armenteros et al., 2017) has become a standard in the field. Such static standards facilitate method comparisons. To further probe results, we created a new test set (*setHARD*), which was redundancy-reduced both with respect to itself and all proteins in the *DeepLoc* set (comprised of training data and *setDeepLoc*, used for testing). For this set, the 10-state accuracy (Q10) dropped, on average, 22 percentage points with respect to the static standard (Figure 2). We argue that this large margin may be attributed to some combination of the following coupled effects.

(1) All new methods may simply have been substantially overfitted to the static data set, e.g., by misusing the test set for hyperparameter optimization. This could partially explain the increase in performance on *setHARD* when mimicking the class distributions in the training set and *setDeepLoc*.

(2) The static standard set allowed for some level of sequence-redundancy (information leakage) at various levels: certainly within the test set, which had not been redundancy reduced to itself, maybe also between the training and test set. Methods with many free parameters might more easily zoom into exploiting such residual sequence similarity for prediction because proteins with similar sequence locate in similar compartments. In fact, this may explain the somewhat surprising observation that *DeepLoc* appeared to perform worse on *setHARD* using evolutionary information

instead of a generic BLOSUM metric (Figure 2: *DeepLoc62* vs. *DeepLoc*). Residual redundancy is much easier to capture via evolutionary information than by BLOSUM (Urban et al., 2020) (for computational biologists: the same way in which PSI-BLAST (Altschul et al., 1997) outperforms pairwise BLAST).

(3) The confusion matrix (Figure 3) demonstrated how classes with more experimental data tended to be predicted more accurately. As *setDeepLoc* and *setHARD* differed in their class composition, even without overfitting and redundancy, prediction methods would perform differently on the two. In fact, this can be investigated by recomputing the performance on a similar class-distributed superset of *setHARD*, on which performance dropped only by 11, 24, 18, and 17 percentage points for *DeepLoc62*, *DeepLoc*, *LA ProtBert*, and *LA ProtT5*, respectively.

Overall, several overlaying effects caused the performance to drop between the two data sets. Interestingly, different approaches behaved alike: both for alternative inputs from protein language models (*SeqVec*, *ProtBert*, *ProtT5*) and for alternative methods (AT, MLP, LA), of which one (AT) refrained from weight optimization.

What can users expect from subcellular location predictions? If the top accuracy for one data set was Q10 $\sim 60\%$ and Q10 $\sim 80\%$ for the other, what can users expect for their next ten queries: six correct or eight, or 6-8? The answer depends on the query: if those proteins were sequence similar to proteins with known location (case: redundant): the answer is eight. Conversely, for new proteins (without homologs of known location), six in ten will be correctly predicted, on average. In turn, this implies that for novel proteins, there seems to be significant room for pushing performance to further heights, possibly by combining *LA ProtBert*/*LA ProtT5* with evolutionary information.

6. Conclusion

We presented a light attention mechanism (LA) in an architecture operating on language model embeddings of protein sequences, namely those from *SeqVec* (Heinzinger et al., 2019), *ProtBert*, and *ProtT5* (Elnaggar et al., 2020). LA efficiently aggregated information and coped with arbitrary sequence lengths, thereby mastering the enormous range of proteins spanning from 30-30 000 residues. By implicitly assigning a different importance score for each sequence position, the method succeeded in predicting protein subcellular location much better than methods based on simple pooling. More importantly, for two protein LMs, LA succeeded in outperforming the state-of-the-art without using evolutionary-based inputs, i.e., the single most important input feature for previous methods. This constituted an important breakthrough: although many methods had come close to the state-of-the-art using embeddings instead of evolutionary information, none had ever overtaken as the methods presented here. Our best method was based on the largest protein LM, namely on *ProtT5* (*LA ProtT5* in Figure 2). Many location prediction methods have been assessed on a standard data set (here: *setDeepLoc*) introduced a few years ago (Almagro Armenteros et al., 2017). Using a new, more challenging data set (*setHARD*), the performance of all methods appeared to drop by around 22 percentage points. While class distributions and data set redundancy (or homology) may explain some of this drop, over-fitting might have also contributed. Overall, the drop underlined that many challenges remain to be addressed by future methods. For the time being, the best methods *LA ProtBert* and *LA ProtT5*, are freely available via a webserver (embed.protein.properties) and as part of a high-throughput pipeline (Dallago et al., 2021).

Acknowledgements

Thanks to Tim Karl (TUM) for help with hardware and software; to Inga Weise (TUM) for support with many other aspects of this work. Thanks to the Rostlab for constructive conversations. Thanks to all those who deposit their experimental data in public databases, and to those who maintain these databases. In particular, thanks to the Ioanis Xenarios (SIB, Univ. Lausanne), Matthias Uhlen (Univ. Upssala) and their teams at Swiss-Prot and HPA. This work was supported by the Deutsche Forschungsgemeinschaft (DFG) – project number RO1320/4-1, by the Bundesministerium für Bildung und Forschung (BMBF) – project number 031L0168, and by the BMBF through the program "Software Campus 2.0 (TU München)" – project number 01IS17049.

References

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, December 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0598-1. URL https://www.nature.com/articles/s41592-019-0598-1. Number: 12 Publisher: Nature Publishing Group.

Almagro Armenteros, J. J., Sønderby, C. K., Sønderby, S. K., Nielsen, H., and Winther, O. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, November 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx431. URL https://academic.oup.com/bioinformatics/article/33/21/3387/3931857. tex.ids: almagroarmenterosDeepLocPredictionProtein2017a publisher: Oxford Academic.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25 (17):3389–3402, September 1997. ISSN 0305-1048. doi: 10.1093/nar/25.17.3389. URL https://doi.org/10.1093/nar/25.17.3389.

Bahdanau, D., Cho, K., and Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The Protein Data Bank. *Nucleic Acids Research*, 28(1): 235–242, January 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235. URL https://doi.org/10.1093/nar/28.1.235.

Bhattacharya, N., Thomas, N., Rao, R., Dauparas, J., Koo, P. K., Baker, D., Song, Y. S., and Ovchinnikov, S. Single Layers of Attention Suffice to Predict Protein Contacts. *bioRxiv*, pp. 2020.12.21.423882, December 2020. doi: 10.1101/2020.12.21.423882. URL https://www.biorxiv.org/content/10.1101/2020.12.21.423882v2. Publisher: Cold Spring Harbor Laboratory Section: New Results.

Blum, T., Briesemeister, S., and Kohlbacher, O. MultiLoc2: integrating phylogeny and Gene Ontology terms improves subcellular protein localization prediction. *BMC bioinformatics*, 10(1):274, 2009. Publisher: Springer.

Briesemeister, S., Blum, T., Brady, S., Lam, Y., Kohlbacher, O., and Shatkay, H. SherLoc2: a high-accuracy hybrid method for predicting subcellular localization of proteins. *Journal of proteome research*, 8(11):5363–5366, 2009. Publisher: ACS Publications.

Briesemeister, S., Rahnenführer, J., and Kohlbacher, O. YLoc—an interpretable web server for predicting subcellular localization. *Nucleic acids research*, 38(suppl_2): W497–W502, 2010. Publisher: Oxford University Press.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., Xie, W., Rosen, G. L., Lengerich, B. J., Israeli, J., Lanchantin, J., Woloszynek, S., Carpenter, A. E., Shrikumar, A., Xu, J., Cofer, E. M., Lavender, C. A., Turaga, S. C., Alexandari, A. M., Lu, Z., Harris, D. J., DeCaprio, D., Qi, Y., Kundaje, A., Peng, Y., Wiley, L. K., Segler, M. H. S., Boca, S. M., Swamidass, S. J., Huang, A., Gitter, A., and Greene, C. S. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141): 20170387, April 2018. doi: 10.1098/rsif.2017.0387. URL https://royalsocietypublishing.org/doi/10.1098/rsif.2017.0387. Publisher: Royal Society.

Chou, K.-C., Wu, Z.-C., and Xiao, X. iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. *PloS one*, 6(3):e18258, 2011. Publisher: Public Library of Science.

Consortium, T. U. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 2021. doi: 10.1093/nar/gkaa1100. URL https://academic.oup.com/nar/advance-article/doi/10.1093/nar/gkaa1100/6006196.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00994018. URL http://link.springer.com/10.1007/BF00994018.

Dallago, C., Schütze, K., Heinzinger, M., Olenyi, T., Littmann, M., Lu, A. X., Yang, K. K., Min, S., Yoon, S., Morton, J. T., and Rost, B. Learned embeddings from deep learning to visualize and predict protein sets. *Current Protocols in Bioinformatics*, 2021. doi: 10.1002/cpz1.113.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. *bioRxiv*, pp. 2020.07.12.199554, July 2020. doi: 10.1101/2020.07.12.199554. URL https://www.biorxiv.org/content/10.1101/2020.07.12.199554v2. tex.ids: elnaggarProtTransCrackingLanguage2020a publisher: Cold Spring Harbor Laboratory section: New Results.

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

Goldberg, T., Hecht, M., Hamp, T., Karl, T., Yachdav, G., Ahmed, N., Altermann, U., Angerer, P., Ansorge, S., Balasz, K., Bernhofer, M., Betz, A., Cizmadija, L., Do, K. T., Gerke, J., Greil, R., Joerdens, V., Hastreiter, M., Hembach, K., Herzog, M., Kalemanov, M., Kluge, M., Meier, A., Nasir, H., Neumaier, U., Prade, V., Reeb, J., Sorokoumov, A., Troshani, I., Vorberg, S., Waldraff, S., Zierer, J., Nielsen, H., and Rost, B. LocTree3 prediction of localization. *Nucleic Acids Research*, 42(W1): W350–W355, 2014. ISSN 0305-1048. doi: 10.1093/nar/gku396. URL https://doi.org/10.1093/nar/gku396. _eprint: https://academic.oup.com/nar/article-pdf/42/W1/W350/17423232/gku396.pdf.

Gorodkin, J. Comparing two K-category assignments by a K-category correlation coefficient. *Computational Biology and Chemistry*, 28(5): 367 – 374, 2004. ISSN 1476-9271. doi: https://doi.org/10.1016/j.compbiolchem.2004.09.006. URL http://www.sciencedirect.com/science/article/pii/S1476927104000799.

Gribskov, M., McLachlan, A. D., and Eisenberg, D. Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, 84(13): 4355–4358, July 1987. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.84.13.4355. URL https://www.pnas.org/content/84/13/4355. tex.ids= gribskovProfileAnalysisDetection1987a publisher: National Academy of Sciences section: Research Article.

Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., and Rost, B. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, 20(1):723, December 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-3220-8. URL https://doi.org/10.1186/s12859-019-3220-8. tex.ids: heinzingerModelingAspectsLanguage2019a.

Henikoff, S. and Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, November 1992. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.89.22.10915. URL http://www.pnas.org/cgi/doi/10.1073/pnas.89.22.10915.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735. _eprint: https://doi.org/10.1162/neco.1997.9.8.1735.

Horton, P., Park, K.-J., Obayashi, T., Fujita, N., Harada, H., Adams-Collier, C., and Nakai, K. WoLF PSORT: protein localization predictor. *Nucleic acids research*, 35 (suppl_2):W585–W587, 2007. Publisher: Oxford University Press.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Lange, A., Mills, R. E., Lange, C. J., Stewart, M., Devine, S. E., and Corbett, A. H. Classical Nuclear Localization Signals: Definition, Function, and Interaction with Importin alpha,. *Journal of Biological Chemistry*, 282(8):5101–5105, February 2007. ISSN 0021-9258. doi: 10.1074/jbc.R600026200. URL http://www.sciencedirect.com/science/article/pii/S0021925820688019.

Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T., and Rost, B. Embeddings from deep learning transfer GO annotations beyond homology. *Scientific Reports*, 11(1): 1160, January 2021. ISSN 2045-2322. doi: 10.1038/s41598-020-80786-0. URL https://www.nature.com/articles/s41598-020-80786-0. Number: 1 Publisher: Nature Publishing Group.

Mahlich, Y., Steinegger, M., Rost, B., and Bromberg, Y. HFSP: high speed homology-driven function annotation of proteins. *Bioinformatics*, 34(13):i304–i312, July 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty262. URL https://doi.org/10.1093/bioinformatics/bty262.

McInnes, L., Healy, J., Saul, N., and Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.*, 3(29):861, 2018. doi: 10.21105/joss.00861. URL https://doi.org/10.21105/joss.00861.

Nair, R. and Rost, B. Mimicking Cellular Sorting Improves Prediction of Subcellular Localization. *Journal of Molecular Biology*, 348(1):85–100, April 2005. ISSN 0022-2836. doi: 10.1016/j.jmb.2005.02.025. URL http://www.sciencedirect.com/science/article/pii/S0022283605001774.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Associa-

tion for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Pierleoni, A., Martelli, P. L., Fariselli, P., and Casadio, R. BaCelLo: a balanced subcellular localization predictor. *Bioinformatics*, 22(14):e408–416, July 2006.

Pierleoni, A., Martelli, P. L., and Casadio, R. MemLoci: predicting subcellular localization of membrane proteins in eukaryotes. *Bioinformatics*, 27(9):1224–1230, May 2011.

Radford, A. Improving Language Understanding by Generative Pre-Training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language Models are Unsupervised Multitask Learners. 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. Evaluating Protein Transfer Learning with TAPE. *Advances in neural information processing systems*, 32:9689–9701, December 2019. ISSN 1049-5258. URL https://pubmed.ncbi.nlm.nih.gov/33390682.

Rao, R., Ovchinnikov, S., Meier, J., Rives, A., and Sercu, T. Transformer protein language models are unsupervised structure learners. *bioRxiv*, pp. 2020.12.15.422761, December 2020. doi: 10.1101/2020.12.15.422761. URL https://www.biorxiv.org/content/10.1101/2020.12.15.422761v1. tex.ids: raoTransformerProteinLanguage2020a publisher: Cold Spring Harbor Laboratory section: New Results.

Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences. *bioRxiv*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/early/2019/04/29/622803.

Rost, B. and Sander, C. Prediction of protein secondary structure at better than 70% accuracy. *Journal of molecular biology*, 232(2):584–599, 1993. tex.ids: RN1 publisher: Elsevier Science type: Journal article.

Rost, B., Liu, J., Nair, R., Wrzeszczynski, K. O., and Ofran, Y. Automatic prediction of protein function. *Cellular and Molecular Life Sciences*, 60(12):2637–2650, 2003. URL http://www.rostlab.org/papers/2003_rev_func/. Type: Journal article.

Rui Kuang, Ie, E., Ke Wang, Kai Wang, Siddiqi, M., Freund, Y., and Leslie, C. Profile-based string kernels for remote homology detection and motif extraction. In *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004.*, pp. 146–154, Stanford, CA, USA, 2004. IEEE. ISBN 978-0-7695-2194-7. doi: 10.1109/CSB.2004.1332428. URL http://ieeexplore.ieee.org/document/1332428/.

Savojardo, C., Martelli, P. L., Fariselli, P., and Casadio, R. SChloro: directing Viridiplantae proteins to six chloroplastic sub-compartments. *Bioinformatics*, 33(3):347–353, 2017.

Savojardo, C., Martelli, P. L., Fariselli, P., Profiti, G., and Casadio, R. BUSCA: an integrative web server to predict subcellular localization of proteins. *Nucleic Acids Research*, 46(W1):W459–W466, 2018. ISSN 0305-1048. doi: 10.1093/nar/gky320. URL https://doi.org/10.1093/nar/gky320. _eprint: https://academic.oup.com/nar/article-pdf/46/W1/W459/25110557/gky320.pdf.

Schuster, M. and Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093. URL https://doi.org/10.1109/78.650093.

Steinegger, M. and Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, November 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988. URL https://doi.org/10.1038/nbt.3988.

Steinegger, M. and Söding, J. Clustering huge protein sequence sets in linear time. *Nature Communications*, 9(1):2542, June 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-04964-5. URL https://doi.org/10.1038/s41467-018-04964-5.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and the UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, March 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu739. URL https://doi.org/10.1093/bioinformatics/btu739.

Urban, G., Torrisi, M., Magnan, C. N., Pollastri, G., and Baldi, P. Protein profiles: Biases and protocols. *Computational and Structural Biotechnology Journal*, 18:2281 – 2289, 2020. ISSN 2001-0370. doi: https://doi.org/10.1016/j.csbj.2020.08.015. URL

http://www.sciencedirect.com/science/article/pii/S2001037020303688. tex.ids: urbanProteinProfilesBiases2020a.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Yu, C.-S., Chen, Y.-C., Lu, C.-H., and Hwang, J.-K. Prediction of protein subcellular localization. *Proteins: Structure, Function, and Bioinformatics*, 64(3):643–651, 2006. tex.ids: yuPredictionProteinSubcellular2006a publisher: Wiley Online Library.

Appendix: Light Attention Predicts Protein Location from the Language of Life

Hannes Stärk [*1] **Christian Dallago** [*1 2] **Michael Heinzinger** [1 2] **Burkhard Rost** [1 3 4]

1. Protein Preliminaries

Protein Sequences. Proteins are built by chaining and arbitrary number of one of 20 amino acids in a particular order. When amino acids come together to form protein sequences, they are dubbed residues. During the assembly in the cell, constrained by physiochemical forces, the one-dimensional chains of residues fold into unique 3D shapes based solely on their sequence that largely determine protein function. The ideal machine learning model would predict a protein's 3D shape and thus function from just protein sequence (the ordered chain of residues).

Protein Subcellular Location. Eukaryotic cells contain different organelles/compartments. Each organelle serves a purpose, e.g., ribosomes chain together new proteins while mitochondria synthesize ATP. Proteins are the machinery used to perform these functions, including transport in and out and communication between different organelles and a cell's environment. For some compartments, e.g., the nucleus, special stretches of amino acids, e.g., nuclear localization signals (NLS), help identifying a protein's location via simple string matching. However, for many others, the localization signal is diluted within the whole sequence, requiring sequence-level predictions. Furthermore, some organelles (and the cell itself) feature membranes with different biochemical properties than the inside or outside, requiring protein gateways.

Homology-inference. Two highly similar protein sequences will most likely fold in similar 3D structures and more likely to perform similar functions. Homology based inference (Nair & Rost, 2002; Mahlich et al., 2018), which transfers annotations of experimentally validated proteins to query protein sequences, is based on this assumption (Sander & Schneider, 1991). Practically this means searching a database of annotated protein sequences for sequences that meet both an identity threshold and a length-of-match threshold to some query protein sequence. Sequence homology delivers good results, but its stringent requirements render it applicable to only a fraction of proteins (Rost, 1999).

Machine learning Function Prediction. When moving into territory where sequence similarity is less conserved for shorter stretches of matching sequences (Mahlich et al., 2018; Rost, 2002), one can try predicting function using evolutionary information and machine learning (Goldberg et al., 2012; Almagro Armenteros et al., 2017). Evolutionary information from protein profiles, encoding a protein's evolutionary path, is obtained by aligning sequences from a protein database to a query protein sequence and computing conservation metrics at the residue level. Using profiles leads to impressively more accurate predictions for sequences with no close homologs and has been the standard for most protein prediction tasks (Urban et al., 2020), including subcellular localization (Goldberg et al., 2012; Almagro Armenteros et al., 2017; Savojardo et al., 2018). While profiles provide a strong and useful inductive bias, their information content heavily depends on a balance of the number of similar proteins (depth), the overall length of the matches (sequence coverage), the diversity of the matches (column coverage), and their generation is parameter sensitive.

2. Hyperparameters

The following describes the search space used to find hyperparameters of our final LA and FFN models. We performed random search over these parameters. The evaluated learning rates were in the range of $[5 \times 10^{-6}$ - $5 \times 10^{-3}]$. For the light attention architecture, we tried filter sizes $[3, 5, 7, 9, 11, 13, 15, 21]$ and hidden sizes $d_{out} \in [32, 128, 256, 512, 1024, 1500, 2048]$, as well as concatenating outputs of convolutions with different filter sizes. For the FFN, we searched over the hidden layer sizes $[16, 32, 64, 512, 1024]$, where 32 was the optimium. We maximized batch size to fit a Quadro RTX 8000 with 48GB vRAM, resulting in the batch size of 150. Note that the memory requirement is dependent on the size of the longest sequence in a batch. In the DeepLoc dataset, the longest sequence had 13 100 residues.

3. Additional Results

We provide results for both *setDeepLoc* (Table 4) and *setHARD* (Table 3) in tabular form, including the Matthew's Correlation Coefficients (MCC) and class unweighted F1 score.

Table 1. MCC of additional baselines and ablations compared to the LA architecture on *ProtT5* embeddings (above the line) of *setDeepLoc* and *setHARD* averaged over 10 seeds. The best method is **bold** and the second best is underlined.

METHOD	SETDEEPLOC	SETHARD
LA PROTT5	**.831**± .004	**.577**± .007
LA - SOFTMAX	.828± .004	.570± .008
LA - MAXPOOL	.816± .002	.559± .008
ATTENTION FROM V	.824± .003	.571± .012
DEEPLOC LSTM	.752± .010	.505± .009
CONV + ADAPOOL	.785± .010	.526± .022
MEANPOOL + FFN	.785± .006	.529± .010
LA ON ONEHOT	.326± .012	.216± .014
LA ON PROFILES	.302± .016	.195± .022

Table 2. Class unweighted F1 score of additional baselines and ablations compared to the LA architecture on *ProtT5* embeddings (above the line) of *setDeepLoc* and *setHARD* averaged over 10 seeds. The best method is **bold** and the second best is underlined.

METHOD	SETDEEPLOC	SETHARD
LA PROTT5	**.854**± .004	**.642**± .004
LA - SOFTMAX	.850± .004	.633± .008
LA - MAXPOOL	.842± .002	.632± .006
ATTENTION FROM V	.845± .004	.634± .011
DEEPLOC LSTM	.788± .009	.590± .007
CONV + ADAPOOL	.818± .010	.608± .020
MEANPOOL + FFN	.814± .005	.604± .008
LA ON ONEHOT	.367± .025	.262± .033
LA ON PROFILES	.384± .018	.279± .019

Table 3. Accuracy and Matthew's correlation coefficient (MCC) on *setHARD*.

METHOD	ACCURACY	MCC
DEEPLOC62	56.94	0.476
DEEPLOC	51.36	0.410
AT PROTBERT	42.04	0.306
AT PROTT5	55.01	0.454
FFN PROTBERT	53.16± 1.19	0.429± 0.014
FFN PROTT5	61.27± 0.98	0.529± 0.011
LA PROTBERT	58.36± 1.02	0.490± 0.012
LA PROTT5	**65.21**± 0.61	**0.577**± 0.007

Table 4. Accuracy and Matthew's correlation coefficient (MCC) on *setDeepLoc*.

METHOD	ACCURACY	MCC
LOCTREE2	61.20	0.525
MULTILOC2	55.92	0.487
SHERLOC2	58.15	0.511
YLOC	61.22	0.533
CELLO	55.21	0.454
ILOC-EUK	68.20	0.641
WOLF PSORT	56.71	0.479
DEEPLOC62	73.60	0.683
DEEPLOC	77.97	0.735
AT SEQVEC	60.97	0.508
AT PROTBERT	64.85	0.567
AT PROTT5	73.92	0.687
FFN SEQVEC	70.57± 0.93	0.636± 0.011
FFN PROTBERT	75.88± 0.45	0.702± 0.006
FFN PROTT5	82.28± 0.51	0.786± 0.006
LA SEQVEC	75.63± 0.11	0.705± 0.002
LA PROTBERT	80.29± 0.21	0.762± 0.002
LA PROTT5	**86.01**± 0.34	**0.832**± 0.004

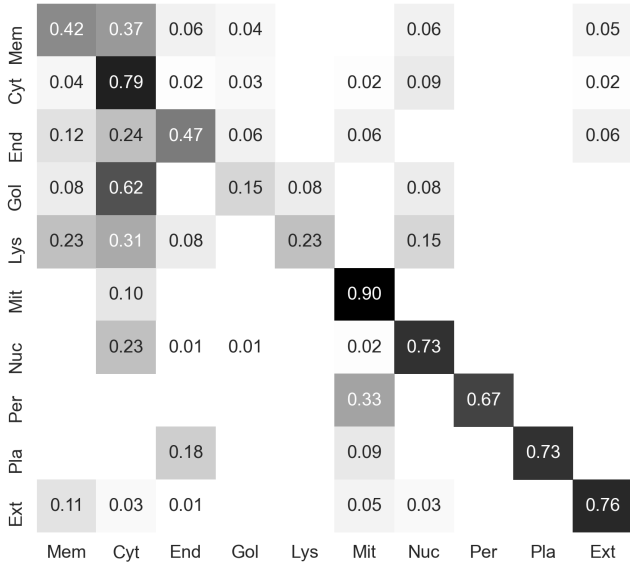

Figure 1. Confusion matrix of LA predictions on ProtT5 (Elnaggar et al., 2020) embeddings for *setHARD* annotated with the fraction of the true class. Vertical axis: true class, horizontal axis: predicted class. Labels: Mem=cell **Mem**brane; Cyt=**Cyt**oplasm; End=**End**oplasmatic Reticulum; Gol=**Gol**gi apparatus; Lys=**Lys**osome/vacuole; Mit=**Mit**ochondrion; Nuc=**Nuc**leus; Per=**Per**oxisome; Pla=**Pla**stid; Ext=**Ext**racellular

4. Datasets

Table 5 shows the distribution of subcellular localization classes in the *setDeepLoc* and our new *setHARD*.

Table 5. Number of proteins and percentage of dataset for each class for the DeepLoc dataset and our *setHARD*. ER abbreviates Endoplasmatic Reticulum

LOCATION	DEEPLOC		SETHARD	
	#	%	#	%
NUCLEUS	4043	28.9	99	20.2
CYTOPLASM	2542	19.3	117	23.8
EXTRACELLULAR	1973	14.0	92	18.8
MITOCHONDRION	1510	11.8	10	2.0
CELL MEMBRANE	1340	9.5	98	20.0
ER	862	6.2	34	6.9
PLASTID	757	5.4	11	2.6
GOLGI APPARATUS	356	2.6	13	2.6
LYSOSOME/VACUOLE	321	2.3	13	2.2
PEROXISOME	154	1.1	3	0.6

4.1. New test set creation

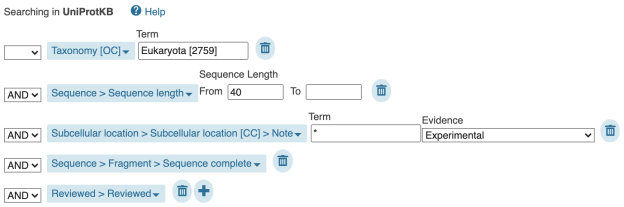

Figure 2. Screenshot of the filtering options applied to the advanced UniProt search (uniprot.org/uniprot).

In the following, we lay out the steps taken to produce the new test set (*setHARD*). The starting point is a filtered UniProt search with options as selected in Figure 2. Python code used is available here: *OMITTED*.

- Download data as FASTA & XML:

```
wget "https://www.uniprot.org/
uniprot/?query=taxonomy:%
22Eukaryota%20[2759]%22%
20length:[40%20TO%20*]%
20locations:(note:*%20evidence:%
22Inferred%20from%20experiment%
20[ECO:0000269]%22)%20fragment:no%
20AND%20reviewed:yesformat=
xmlforce=truesort=scorecompress=
yes"
```

```
wget "https://www.uniprot.org/
uniprot/?query=taxonomy:%
22Eukaryota%20[2759]%22%
20length:[40%20TO%20*]%
20locations:(note:*%20evidence:%
22Inferred%20from%20experiment%
20[ECO:000026%22)%20fragment:no%
20AND%20reviewed:yesformat=
fastaforce=truesort=scorecompress=
yes"
```

- Download deeploc data:

```
wget http://www.cbs.dtu.dk/services/
DeepLoc-1.0/deeploc_data.fasta
```

- Align sequences in swissprot to deeploc that have more than 20% PIDE:

```
mmseqs easy-search swissprot.fasta
deeploc_data.fasta -s 7.5
--min-seq-id 0.2 --format-output
query,target,fident,alnlen,mismatch,
gapopen,qstart,qend,tstart,tend,
evalue,bits,pident,nident,qlen,tlen,
qcov,tcov alignment.m8 tmp
```

- Extract localizations from SwissProt XML:

```
python extract_localizaiotns_from_
swissprot.py
```

- Map deeploc compartments on swissprot localizations & remove duplicates ([P123, Nucleus] appearing twice), remove multilocated ([P123, Nucelus] and [P123, Cytoplasm] –> remove P123) empty or not experimental annotations:

```
python map_and_filter_swissprot_
annotations.py
```

- Create FASTA like deeploc from sequences not in alignment:

```
python extract_unaligned_
sequences.py
```

- Redundancy reduce new set to 20%:

```
mmseqs easy-cluster --min-seq-id
0.2 new_test_set_not_redundancy_
reduced.fasta new_hard_test_set_
PIDE20.fasta tmp
```

References

Almagro Armenteros, J. J., Sønderby, C. K., Sønderby, S. K., Nielsen, H., and Winther, O. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, November 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx431. URL https://academic.oup.com/bioinformatics/

article/33/21/3387/3931857. tex.ids: almagroarmenterosDeepLocPredictionProtein2017a publisher: Oxford Academic.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. *bioRxiv*, pp. 2020.07.12.199554, July 2020. doi: 10.1101/2020.07.12.199554. URL https://www.biorxiv.org/content/10.1101/2020.07.12.199554v2. tex.ids: elnaggarProtTransCrackingLanguage2020a publisher: Cold Spring Harbor Laboratory section: New Results.

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

Mahlich, Y., Steinegger, M., Rost, B., and Bromberg, Y. HFSP: high speed homology-driven function annotation of proteins. *Bioinformatics*, 34(13):i304–i312, July 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty262. URL https://doi.org/10.1093/bioinformatics/bty262.

Nair, R. and Rost, B. Sequence conserved for subcellular localization. *Protein Science*, 11(12):2836–2847, 2002. ISSN 1469-896X. doi: https://doi.org/10.1110/ps.0207402. URL https://onlinelibrary.wiley.com/doi/abs/10.1110/ps.0207402.

Rost, B. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, 12(2):85–94, February 1999. ISSN 1741-0126. doi: 10.1093/protein/12.2.85. URL https://doi.org/10.1093/protein/12.2.85.

Rost, B. Enzyme Function Less Conserved than Anticipated. *Journal of Molecular Biology*, 318 (2):595–608, April 2002. ISSN 0022-2836. doi: 10.1016/S0022-2836(02)00016-5. URL http://www.sciencedirect.com/science/article/pii/S0022283602000165.

Sander, C. and Schneider, R. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Bioinformatics*, 9(1):56–68, 1991. ISSN 1097-0134. doi: https://doi.org/10.1002/prot.340090107. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.340090107.

Savojardo, C., Martelli, P. L., Fariselli, P., Profiti, G., and Casadio, R. BUSCA: an integrative web server to predict subcellular localization of proteins. *Nucleic Acids Research*, 46(W1):W459–W466, 2018. ISSN 0305-1048. doi: 10.1093/nar/gky320. URL https://doi.org/10.1093/nar/gky320. _eprint: https://academic.oup.com/nar/article-pdf/46/W1/W459/25110557/gky320.pdf.

Urban, G., Torrisi, M., Magnan, C. N., Pollastri, G., and Baldi, P. Protein profiles: Biases and protocols. *Computational and Structural Biotechnology Journal*, 18:2281 – 2289, 2020. ISSN 2001-0370. doi: https://doi.org/10.1016/j.csbj.2020.08.015. URL http://www.sciencedirect.com/science/article/pii/S2001037020303688. tex.ids: urbanProteinProfilesBiases2020a.