LIGHT ATTENTION PREDICTS PROTEIN LOCATION FROM THE LANGUAGE OF LIFE

Anonymous authors
Paper under double-blind review

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 three to five percentage points (Q10). The novel models are available as a web-service and as a stand-alone application at OMITTED.com.

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).

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, 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

Data. 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 dataset contained 13 858 proteins annotated with experimental evidence for one of ten location classes. 2 768 proteins made up the test set (henceforth called *setDeepLoc*). 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 *setHARD*. It contains more difficult targets, has more stringent redundancy reduction, and a different class distribution than the DeepLoc training data. Details on the datasets are provided in the appendix.

Model Input: protein embeddings. As input to the LA architectures, we extracted embeddings from three pre-trained protein language models (LMs): the bidirectional LSTM SeqVec (Heinzinger et al., 2019) based on ELMo (Peters et al., 2018), the encoder-only model ProtBert (Elnaggar et al., 2020) based on BERT (Devlin et al., 2019), and the encoder-only model ProtT5 (Elnaggar et al., 2020) based on T5 (Raffel et al., 2020). We obtained embeddings for each residue (NLP equivalent: word) in a protein sequence (NLP equivalent: document) using the bio-embeddings software (Dallago et al., 2020). 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. With a hidden size of 1024 for each LM, inputs to LA 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) classifiers was a protein embedding $\boldsymbol{X} \in \mathbb{R}^{1024 \times L}$. The input was transformed by two separate 1D convolutions with filter sizes s parameterized by learned weights $\mathbf{W}^{(e)}, \mathbf{W}^{(v)} \in \mathbb{R}^{s \times 1024 \times d_{out}}$. The convolutions were applied over the length dimension to produce attention coefficients and value features $\boldsymbol{E}, \boldsymbol{V} \in \mathbb{R}^{d_{out} \times L}$. To use the coefficients as attention distribution over all j, we softmax-normalized over protein length. The attention weight $\boldsymbol{A}_{i,j} \in \mathbb{R}$ for the j-th residue and the i-th feature dimension was calculated as:

$$\boldsymbol{A}_{i,j} = \frac{exp(\boldsymbol{E}_{i,j})}{\sum_{l=1}^{L} exp(\boldsymbol{E}_{i,l})} \tag{1}$$

As the weight distributions for each feature dimension i are independent, they might generate different attention patterns. We used the normalized attention distributions to compute weighted sums over the transformed residue embeddings $\boldsymbol{V}_{i,j}$. Thus, we obtained a fixed-size representation $\boldsymbol{x'} \in \mathbb{R}^{d_{out}}$ for the whole protein, independent of its length.

$$\boldsymbol{x}'_i = \sum_{j=1}^{L} \boldsymbol{A}_{i,j} \boldsymbol{V}_{i,j} \tag{2}$$

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

Methods used for comparison. For comparison, we trained a two layer feed-forward network (FFN) proposed previously (Heinzinger et al., 2019). Instead of per-residue embeddings in $\mathbb{R}^{1024\times L}$, the FFNs 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). Following 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).

4 RESULTS

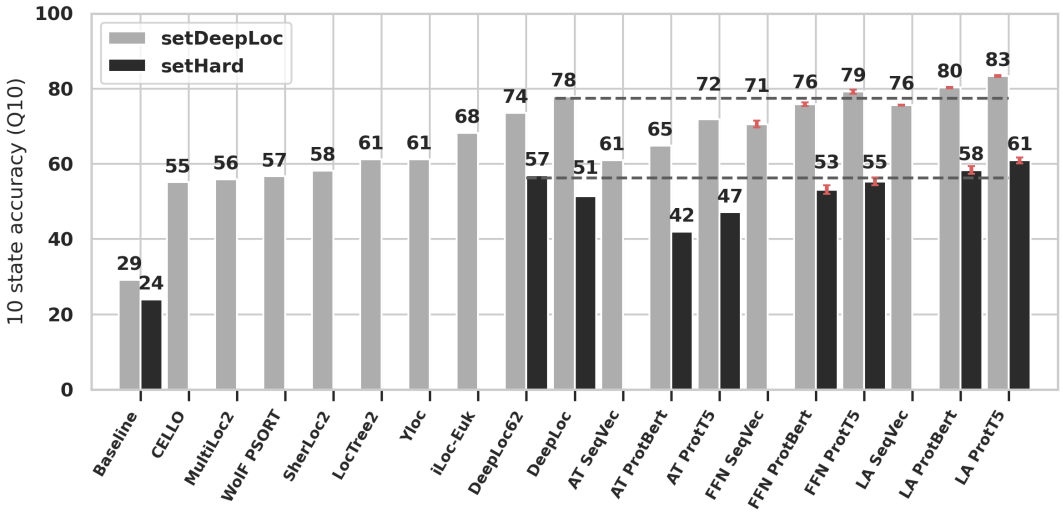

Figure 1: **LA architectures perform best.** Bars give the ten-class accuracy (Q10) for popular location prediction methods on *setDeepLoc* (light-gray bars) and *setHARD* (dark-gray bars). Baseline is the most common class in each set. Horizontal gray dashed lines mark the previous SOTA on either set. 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* (Methods for detail on the external methods used; tabular data in *Appendix: Additional Results*).

Our results show that LM embedding based methods can outperform representations with evolutionary information. The simple AT approach already outperformed some methods that use evolutionary information. Using ProtT5 embeddings, LA improves upon the state-of-the-art (SOTA) (Almagro Armenteros et al., 2017) by 3 and 5 percentage points on *setHARD* and *setDeepLoc*.

Light attention (LA) mechanism crucial. To further evaluate the effectiveness of the LA architecture's aggregation mechanisms, we replaced the light attention that produced x' with averaging the coefficient features e over the length dimension. Performance using ProtT5 embeddings dropped from 83.37% to $81.54 \pm 0.13\%$ (Q10(*setDeepLoc*)). Similarly, we dropped the max-pooled values v^{max} as input to the MLP such that only the aggregated light attention features were used. This reduced performance from 83.37% to $82.23 \pm 0.44\%$ (Q10(*setDeepLoc*)).

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. We provide code to reproduce all results[1].

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

[1]OMITTED

such as SeqVec (Heinzinger et al., 2019), ProtBert, or ProtT5 (Elnaggar et al., 2020) to meaningful sequence-embeddings (NLP equivalent: document). Although averaging surpassed evolutionary-information-based methods using simple similarity-based annotation transfer (Figure 1: AT*) and in one instance even SOTA using a simple feed-forward network (Figure 1: *DeepLoc* vs. *FNN 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*, (Almagro Armenteros et al., 2017; Heinzinger et al., 2019)) in a statistically significant manner on two test sets (Figure 1). To the best of our knowledge, this improvement was the first instance that embedding-based transfer learning substantially outperformed AI/ML methods using evolutionary information 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. The embeddings needed as input for the LA models come with three advantages over evolutionary-information-based input essential for methods such as *DeepLoc* (Almagro Armenteros et al., 2017). Chiefly, embeddings can be obtained in far less time than is needed to generate evolutionary information and require fewer compute resources. Even considering 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 to generate evolutionary information input for a large set of 10 000 proteins. The slowest but most informative embedder (ProtT5) is 3x faster, while the second most informative (ProtBert) is 5x faster (Table 3). Additionally, 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 $>$10GB of vRAM. Lastly, extracting evolutionary information 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).

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). By implicitly assigning a different importance score for each sequence position, the method succeeded in predicting protein subcellular location much better than previous methods. On the standard subcellular localization benchmark and on a newly created harder test set, LA outperformed the state-of-the-art without using evolutionary-based inputs. This constituted an important breakthrough since it is the first time embedding-based approaches beat evolutionary information in function-related predictions. The more accurate and less homology dependent localization predictions can help biologists in discovering protein function and in downstream tasks like drug discovery. As such, we make the best methods *LA ProtBert* and *LA ProtT5* freely available as a web-server[2] and as part of a high-throughput pipeline (Dallago et al., 2020).

[2]OMITTED

REFERENCES

Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. 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.

José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. 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.

Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. 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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In Yoshua Bengio and Yann LeCun (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.

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. 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.

Nicholas Bhattacharya, Neil Thomas, Roshan Rao, Justas Dauparas, Peter K. Koo, David Baker, Yun S. Song, and Sergey Ovchinnikov. 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.

Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, Wei Xie, Gail L. Rosen, Benjamin J. Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne E. Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M. Cofer, Christopher A. Lavender, Srinivas C. Turaga, Amr M. Alexandari, Zhiyong Lu, David J. Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura K. Wiley, Marwin H. S. Segler, Simina M. Boca, S. Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey S. Greene. 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.

The UniProt Consortium. 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.

Corinna Cortes and Vladimir Vapnik. 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.

Christian Dallago, Konstantin Schütze, Michael Heinzinger, Tobias Olenyi, and Burkhard Rost. bio_embeddings: python pipeline for fast visualization of protein features extracted by language models. *F1000Research*, 9, August 2020. doi: 10.7490/f1000research.1118163.1. URL https://f1000research.com/posters/9-876.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (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.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. 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.

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

Tatyana Goldberg, Maximilian Hecht, Tobias Hamp, Timothy Karl, Guy Yachdav, Nadeem Ahmed, Uwe Altermann, Philipp Angerer, Sonja Ansorge, Kinga Balasz, Michael Bernhofer, Alexander Betz, Laura Cizmadija, Kieu Trinh Do, Julia Gerke, Robert Greil, Vadim Joerdens, Maximilian Hastreiter, Katharina Hembach, Max Herzog, Maria Kalemanov, Michael Kluge, Alice Meier, Hassan Nasir, Ulrich Neumaier, Verena Prade, Jonas Reeb, Aleksandr Sorokoumov, Ilira Troshani, Susann Vorberg, Sonja Waldraff, Jonas Zierer, Henrik Nielsen, and Burkhard Rost. 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.

Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. 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.

Sepp Hochreiter and Jürgen Schmidhuber. 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.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun (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.

Allison Lange, Ryan E. Mills, Christopher J. Lange, Murray Stewart, Scott E. Devine, and Anita H. Corbett. 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.

Maria Littmann, Michael Heinzinger, Christian Dallago, Tobias Olenyi, and Burkhard Rost. 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.

Yannick Mahlich, Martin Steinegger, Burkhard Rost, and Yana Bromberg. 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.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 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.

Rajesh Nair and Burkhard Rost. 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.

Rajesh Nair and Burkhard Rost. 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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

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

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

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 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.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. 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.

Roshan Rao, Sergey Ovchinnikov, Joshua Meier, Alexander Rives, and Tom Sercu. 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.

Alexander Rives, Siddharth Goyal, Joshua Meier, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. 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.

Burkhard Rost. 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.

Burkhard Rost. 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.

Burkhard Rost and Chris Sander. 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.

Burkhard Rost, Jinfeng Liu, Rajesh Nair, Kazimierz O. Wrzeszczynski, and Yanay Ofran. 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, E. Ie, Ke Wang, Kai Wang, M. Siddiqi, Y. Freund, and C. Leslie. 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/.

Chris Sander and Reinhard Schneider. 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.

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

Castrense Savojardo, Pier Luigi Martelli, Piero Fariselli, Giuseppe Profiti, and Rita Casadio. 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.

Mike Schuster and Kuldip K. Paliwal. 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.

Martin Steinegger and Johannes Söding. 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.

Martin Steinegger and Johannes Söding. 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.

Gregor Urban, Mirko Torrisi, Christophe N. Magnan, Gianluca Pollastri, and Pierre Baldi. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (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.

Chin-Sheng Yu, Yu-Ching Chen, Chih-Hao Lu, and Jenn-Kang Hwang. Prediction of protein subcellular localization. *Proteins: Structure, Function, and Bioinformatics*, 64(3):643–651, 2006. tex.ids: yuPredictionProteinSubcellular2006a publisher: Wiley Online Library.

A Appendix

B 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.

C Implementation details

Training procedure. For the LA architecture, we trained three models, one for each protein embedding (SeqVec, ProtBert and ProtT5) for subsets of the training set. The 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 protein embeddings, 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.

Hyperparameters. We performed random search over the following parameter spaces. 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.

D Additional Results

Low performance for minority classes. The confusion matrix (Figure 2 of predictions for *set-DeepLoc* 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 2: *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).

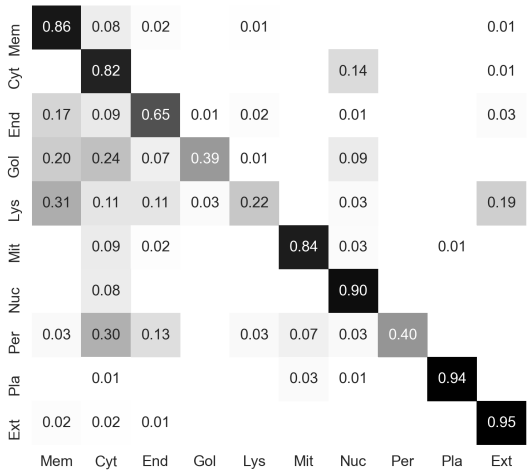

Figure 2: **Mostly capturing majority classes.** Confusion matrix of LA predictions on ProtT5 Elnaggar et al. (2020) embeddings for *setDeepLoc* Almagro Armenteros et al. (2017) annotated with the fraction of the true class. Y-axis (vertical): true class, X-axis (horizontal): predicted class. Labels: 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

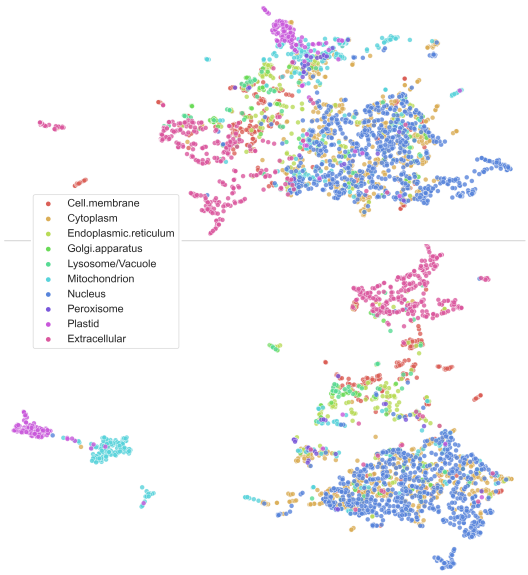

Figure 3: **Qualitative analysis confirms: attention effective.** UMAP McInnes et al. (2018) projections of per-protein embeddings colored according to subcellular location (*setDeepLoc*). Top: ProtT5 embeddings $\boldsymbol{X}$ mean-pooled over protein length (as for FFN/AT input). Bottom: ProtT5 embeddings $\boldsymbol{X}$ weighted according to the attention distribution produced by LA and then summed over the length dimension (this is not $\boldsymbol{x}'$ as we sum the input features $\boldsymbol{X}$ and not the values $\boldsymbol{V}$ after the convolution).

Table 1: 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	<u>77.97</u>	<u>0.735</u>
AT SeqVec	60.97	0.508
AT ProtBert	64.85	0.567
AT ProtT5	71.89	0.661
FFN SeqVec	70.57± 0.93	0.636± 0.011
FFN ProtBert	75.88± 0.45	0.702± 0.006
FFN ProtT5	79.20± 0.55	0.749± 0.007
LA SeqVec	75.63± 0.11	0.705± 0.002
LA ProtBert	80.29± 0.21	0.762± 0.002
LA ProtT5	**83.37**± 0.24	**0.800**± 0.003

Table 2: 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	47.14	0.368
FFN ProtBert	53.16± 1.19	0.429± 0.014
FFN ProtT5	55.31± 1.04	0.457± 0.012
LA ProtBert	58.36± 1.02	0.490± 0.012
LA ProtT5	**60.92**± 0.82	**0.522**± 0.010

We provide results for both *setDeepLoc* (Table 1) and *setHARD* (Table 2) in tabular form, including the Matthew's Correlation Coefficients (MCC). Additionally, Table 3 shows implementation details of the language models to compare their sizes and inference time.

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 1: 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*=66%). *DeepLoc* performed worse on *setHARD* using evolutionary information than using simple sequence information (Figure 1: *DeepLoc* vs. *DeepLoc62*).

Overfitting through standard data set? For protein subcellular location prediction, the data sets from *DeepLoc* (Almagro Armenteros et al., 2017) have 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

Table 3: 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

1). 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 1: *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) 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 ProtT5*, and *LA ProtBert*, 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 (ProtVec, ProtBERT, ProtT5) and for alternative methods (AT, FFN, LA), of which one (AT) refrained from weight optimization.

E DATASETS

The test set *setDeepLoc* has 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.

Novel *setHARD*. 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. Table 4 shows the distribution of subcellular localization classes in the *setDeepLoc* and our new *setHARD*.

Table 4: 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

E.1 New test set creation

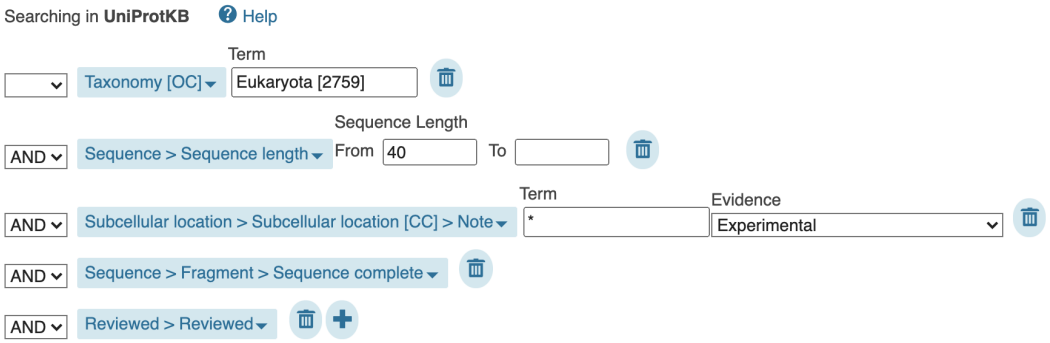

Figure 4: Screenshot of the filtering options applied to the advanced UniProt search (uniprot.org/uniprot).

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 4. 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
```