Opinion
Patterns journal

Transfer learning: the key to functionally annotate the protein universe

L.A. Bugnon, E. Fenoy, A. Edera, J. Raad, G. Stegmayer* and D.H. Milone*

Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL, CONICET, Ciudad Universitaria UNL, (3000) Santa Fe, Argentina. {gstegmayer, dmilone}@sinc.unl.edu.ar

* Corresponding authors

Abstract

The automatic annotation of the protein universe is still an unresolved challenge. Today, there are 229,149,489 entries in the UniProtKB database, but only 0.25% of them have been functionally annotated by expert curators. This manual process integrates knowledge from the protein families database Pfam, where those are annotated with their family domains using sequence alignments and hidden Markov models. This approach has grown the Pfam annotations at a low rate in the last years. Recently, deep learning models appeared with the capability of learning evolutionary patterns from unaligned protein sequences, however requiring large-scale data while many families contain just a few sequences. In this opinion we show how this limitation can be overcome by transfer learning, exploiting the full potential of self-supervised learning on large unannotated data and then supervised learning on a small labeled dataset. We show results where errors in protein family prediction can be reduced by 55% with respect to standard methods.

Introduction

The protein families database (Pfam) is the most widely used repository of protein families and domains. Pfam uses manually curated 'seed' alignments of homologous protein regions (named families) to generate profiles based on hidden Markov models (HMMs). The resulting models are a representation of each profiled family and can be used to classify novel sequences[1]. Even though this approach is very successful, there still remain many proteins of UniProtKB[2] (≈ 25%)

that have not been annotated yet. Moreover, the number of sequences in this knowledge base grows at a much faster rate than its Pfam coverage, introducing novel sequences that may belong to completely new families[3].

Very recently, deep learning (DL) models have emerged[4] to potentially provide a powerful alternative to profile-HMMs which are the dominant technology for protein family classification. DL techniques are capable of inferring patterns shared across the family sequences, allowing autonomous domain annotation on unaligned sequences. This was especially helpful for accelerating the characterization of sequences that do not resemble anything known[5]. However, it is well known that DL techniques rely on large scale data to infer meaningful sequence patterns. This can be a limitation on domain annotation since many Pfam families comprise few seed sequences. Indeed, it has been taken an important step towards overcoming this limitation[4] and we show that this issue can be further significantly reduced with transfer learning (TL) by transferring representations of protein sequences already learned without requiring annotations from large-scale protein data[6].

Transfer learning for protein representations

Transfer learning (Figure 1) is a machine learning technique where one model is first trained with a big unlabeled dataset in a self-supervised way, that is, not using annotations of any specific task, but predicting parts of the same data fed as input (e.g. masked small sub-sequences). This step is also named pre-training, and the result is a task-agnostic deep model and an output model associated with the pretext task for self-supervised learning, which is then discarded. In a second step, the task-agnostic deep model is frozen and what was learned by it is “transferred” to another deep architecture in order to train a new task-specific model. Here another model is trained with supervised learning on a small dataset with labeled data for a specific task (e.g. protein family classification). In summary, TL refers to the situation where what has been learned in one setting is exploited to improve generalization in another one[7]. For proteins there are several already available task-agnostic deep models, which integrate in their output different types of protein information in a compact representation usually named embeddings.

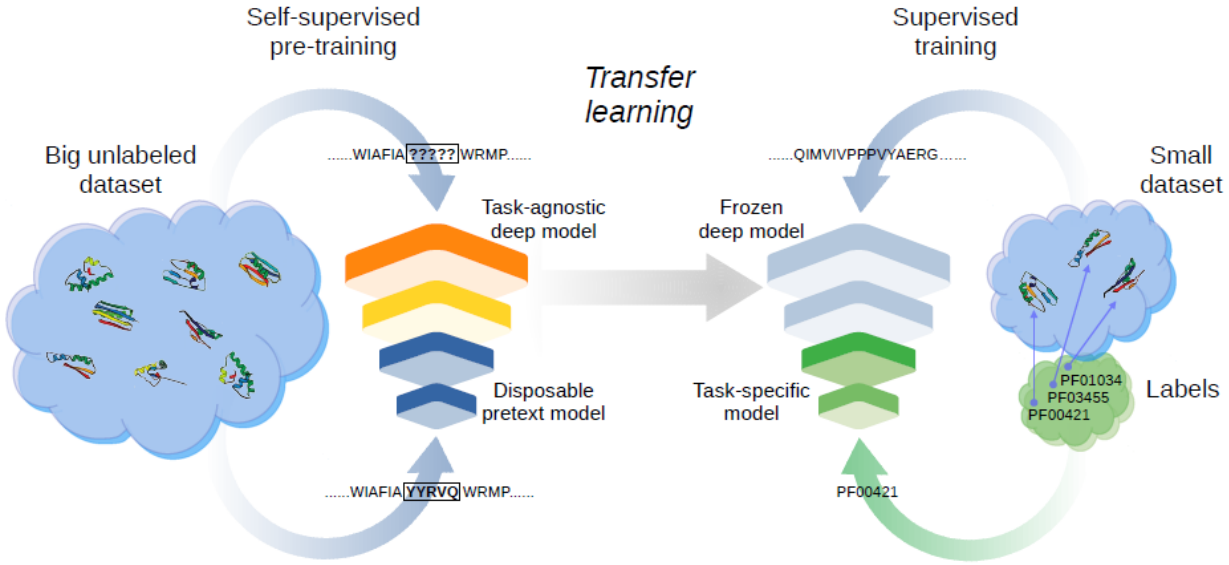

Figure 1: Transfer learning is a machine learning technique where the knowledge gained by training a model on one general task is transferred to be reused in a second specific task. The first model is trained on a big unlabeled dataset, in a self-supervised way (left). This process is known as pre-training, and the result is a task-agnostic deep model (input layers). Through transfer learning, the first layers are frozen (middle) and transferred to another deep architecture. Then, the last layers of the new model are trained with supervised learning on a small dataset with labeled data for a specific task (right).

Protein embeddings are becoming known and required by the community. So much that UniProtKB now provides embeddings as part of the protein annotations (https://www.uniprot.org/help/embeddings). The available protein embeddings were pre-trained on UniRef50, which provides clustered sets of sequences from the complete UniProtKB[2]. A recent review has demonstrated that the Evolutionary Scale Modeling (ESM)[5] is one of the most outstanding protein embeddings in terms of representational power[8]. ESM was trained using 220 million (unaligned) sequences from UniProtKB. ESM is based on Transformers, which have emerged as a powerful general-purpose model architecture for representation learning[9], out-performing deep recurrent and convolutional neural networks. They were originally designed for natural language processing[10], where context within a text is used to predict masked (missing) words. The main hypothesis in this pretext task for self-supervised learning is that the semantics of words can be derived from their contexts. ESM makes an analogy between syllables in text and amino acids in protein sequences: it learns meaningful encodings for each

residue in a self-supervised way, by masking some of the residues in the sequence and trying to predict them. This way, ESM builds an embedding per residue position that encodes the "meaning" of the residue in that context. Then, the per residue representation can be collapsed to a per protein embedding. After this, the ESM learnt representation from UniProtKB, already trained and ready-to-use out of the box, can be "transferred" to be used in a specific downstream task.

Transfer learning for annotating protein domains

In order to illustrate how the use of TL can improve a task like protein domain annotation, we trained a new classifier with Pfam data[4]. Expertly curated sequences from the 17,929 families of Pfam v.32.0 were used to define a benchmark annotation task. Seed sequences from each family were split into challenging train and test sets by clustering them based on sequence similarity. The clustered split provides a benchmark task for annotation of protein sequences with remote homology, that is, sequences in the test set that have low similarity to the ones in the training set. This is useful as an estimation of how well a model will perform with new sequences that are quite different from the ones in the training data. To this end, single-linkage clustering at 25% similarity within each family was used. The resulting benchmark has a distant held-out test set of 21,293 sequences. For this task authors proposed ProtCNN and ProtENN[4]. ProtCNN receives a one-hot coded sequence and learns to automatically extract features to predict family membership. ProtENN is an ensemble of 19 ProtCNNs using a majority vote strategy, where each model was trained with different random parameter initializations.

For the TL approach, we have obtained the ESM embeddings (ESM-1b) of all the train and test sets for the clustered split (a total of 1,339,083 seed sequences). We used two baseline machine learning classifiers for the supervised downstream task: k-nearest neighbor (kNN) and multilayer perceptron (MLP), both trained with the embeddings collapsed to full-sequence, representing each protein domain with a vector in $\mathbb{R}^{1,280}$. After training, these models were tested with the distant held-out test partition for family domain prediction. Finally, we took advantage of TL to improve ProtCNN by training this architecture with the embedding as inputs, instead of the (original) one-hot encoding.

Table 1 shows the results when performance is evaluated by the error rate and the number of errors for classifying the protein domain sequences contained in a held-out clustered test set. The model with the fewest errors is indicated in bold. The first four rows reproduce the ProtCNN, ProtENN, TPHMM and BLASTp results[4]. The next rows show the results obtained when TL is used with different classifiers. The first interesting result is that TL with a simple kNN has obtained at least as good results (27.29% error rate) as ProtCNN (27.60%). Similarly, when transferred to the MLP model or an ensemble of 5 MLPs the error rate is even lower (19.39% and 18.02% respectively). This is a very remarkable result taking into account that embeddings have not been fine-tuned for this particular downstream task. When TL is used as input to a single ProtCNN the results improve even further (15.98%) and the best results are achieved when it is used as the input of an ensemble of 10 TL-ProtCNNs (8.35%). All these cases have achieved better performance than ProtCNN with convolutional feature extraction from a one-hot representation. Moreover, when comparing only ensemble models, the TL-ProtCNN ensemble of 10 models has clearly outperformed the ProtENN ensemble of 19 models (8.35% vs 12.20% error rate, respectively). That is, the error rate has been diminished by 33% thanks to the use of TL for the annotation task. Furthermore, in comparison to the TPHMM the improvement is an impressive 55%.

Table 1: Performance on the clustered split of Pfam.

Method	Error rate (%)	Total errors
ProtCNN	27.60	5,882
ProtENN	12.20	2,590
TPHMM	18.10	3,844
BLASTp	35.90	7,639
TL-kNN	27.29	5,816
TL-MLP	19.39	4,132
TL-MLP-ensemble	18.02	3,840
TL-ProtCNN	15.98	3,405
TL-ProtCNN-ensemble	**8.35**	**1,743**

Closing remarks

In the last few years, several protein representation learning models based on deep learning have

appeared, which provide numerical vectors (embeddings) as a unique representation of the protein. With TL the knowledge encoded in these embeddings can be used in another model to efficiently learn new features of a different downstream prediction task. This TL process allows models to improve their performance by passing knowledge from one task to another, exploiting the information of larger and unlabeled datasets. Protein embeddings has become a new and highly active area of research, with a large number of variants already available in public repositories and easy to use.

The results achieved in a challenging partition of the full Pfam database, with low similarity between train and test sets, have shown superior performance when TL is used in comparison to previous deep learning models. Even in the case of the most simple machine learning classifiers, such as kNN and MLP, the decrease in the error rate was remarkable. Moreover, the best performance is achieved when a convolutional based model is mixed with a pre-trained protein embedding based on transformers. In terms of computational power, even half of ensemble members provided a 33% of improvement in the classification performance.

We hope that this comment will make researchers consider the potential of TL for building better models for protein function prediction. On the practical side, instead of building one's own embedder for proteins, it is very useful to reuse all the computation time already spent building the available learnt representations. Leveraging TL for new tasks with small sets of annotated sequences is easy to implement and provides significant impact on final performance.

Acknowledgments

This work was supported by Agencia Nacional de Promocion Cientifica y Tecnologica (ANPCyT) [PICT 2018 3384, PICT 2018 2905] and UNL [CAI+D 2020 115]. The authors thank and acknowledge Dr. Alex Bateman for his comments on the manuscript.

Author Contributions

Conceptualization, G.S. and D.H.M.; Supervision, G.S.; Methodology, D.H.M.; Investigation and Software, L.A.B., E.F., A.A.E., D.H.M. and J.R.; Writing Original Draft, G.S.; Review and Editing, D.H.M., L.A.B., E.F., A.A.E. and J.R.; Visualization, D.H.M.

Availability

The source code used for this manuscript is freely available in https://github.com/sinc-lab/transfer-learning-pfam

References

1. J. Mistry, P. Coggill, R.Y. Eberhardt, A. Deiana, A. Giansanti, R.R. Finn, A. Bateman, M. Punta. The challenge of increasing pfam coverage of the human proteome. Database, bat040(2013), pp. 1-10, 10.1093/database/bat040.

2. The Uniprot Consortium. UniProt: the universal protein knowledgebase. Nucleic Acids Research, 45(2013), pp. D158–D169.

3. J. Mistry, S. Chuguransky, L. Williams, M. Qureshi, G.A. Salazar, E.L.L. Sonnhammer, S.C.E. Tosatto, L. Paladin, S. Raj, L.J. Richardson, R.D. Finn, A. Bateman. Pfam: The protein families database in 2021. Nucleic Acids Research, 49(2021), pp. D412–D419, 10.1093/nar/gkaa913.

4. M.L. Bileschi, D. Belanger, D.H. Bryant, T. Sanderson, B. Carter, D. Sculley, A. Bateman, M.A. DePristo, L.J. Colwell. Using deep learning to annotate the protein universe. Nature Biotechnology, 40:6(2022), pp. 932–937, 10.1038/s41587-021-01179-w.

5. A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C.L. Zitnick, J. Ma, R. Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proceedings of the National Academy of Sciences, 118:15(2021), pp. e2016239118, 10.1073/pnas.2016239118.

6. S. Unsal, H. Atas, M. Albayrak, K. Turhan, A.C. Acar, T. Dogan. Learning functional properties of proteins with language models. Nature Machine Intelligence, 4:3(2022), pp. 227–245, 10.1038/s42256-022-00457-9.

7. I.J. Goodfellow, Y. Bengio, A. Courville. Deep Learning. MIT Press (2016), Cambridge, MA, USA. http://www.deeplearningbook.org.

8. E. Fenoy, A.A. Edera, G. Stegmayer. Transfer learning in proteins: evaluating novel protein learned representations for bioinformatics tasks. Briefings in Bioinformatics, 23:4(2022), pp. bbac232, 10.1093/bib/bbac232.

9. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, 30 (2017), pp. 5998–6008.

10. J. Devlin, M.W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1 (2019), pp. 4171–4186.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
L. A. Bugnon, E. Fenoy, A. Edera, J. Raad, G. Stegmayer & D. H. Milone; "Transfer learning: The key to functionally annotate the protein universe"
Patterns, 2023.