Learning co-evolution information with natural language processing for protein folding problem*

Egor Zverev[1]***, Sergei Grudinin***[2]***, and Ilia Igashov***[1,2]

zverev.eo@phystech.edu; sergei.grudinin@inria.fr; igashov.is@phystech.edu

[1]Organization, address; [2]Organization, address

Co-evolution information is crucial for building protein sequence descriptors. Its computation is traditionally based on Multiple Sequence Alignments. Applications of MSA are limited, for instance, it fails for sequences with shallow alignment. In this work, we investigate pre-trained language models as a potential alternative to MSA. Specifically, our main focus is fold classification problem. We examine several state-of-the-art methods and modify them by replacing MSA-based feature-generation parts with pre-trained LMs. We compare language models and MSA in their ability to extract sequence evolution information.

Keywords: *protein fold classification; co-evolution, feature generation; transformers; BERT;*

DOI: 10.21469/22233792

1 Introduction

Protein properties are determined by its shape, which is specified by its amino acid sequence [13]. Therefore it is important to learn how to analyze this sequence. Any analysis[...] begins by constructing protein sequence descriptors. Their essential part is a co-evolution information. This information is obtained by searching for homologues of the protein in large databases and further computing multiple sequence alignment (MSA) on it. This technique requires significant computational efforts, moreover, it does not guarantee precise results since it fully relies on finite databases. Besides, this method fails for sequences with shallow alignment

Co-evolution-based methods assume that a mutation of one amino acid leads to mutations of others. Therefore, given an amino acid sequence it is natural to look for other similar sequences. That is performed with MSA [5]. This method fails when we are dealing with the sequences that did not evolve a lot. They have shallow alignments.There are only a few similar sequences in the database. Therefore, information extracted from these alignments is insignificant.

Computing MSA is a heavy task. Alignment databases are finite. If for a given sequence there are no matches in the database, MSA-based methods fail to produce reliable results.

In this work, we analyse pre-trained language models (LMs) [8] as a potential alternative to traditional MSA approaches. It is assumed that protein sequences are not random [4], that

*

there is logic in amino acids structure. Therefore, a set of all amino acids could be seen as a language with complicated inner rules. With the invention of BERT [6] it has become possible to learn the structure of any abstract language. It is natural to assume that by learning amino acids structure, BERT will be able to implicitly learn co-evolution information.

Our main aim is to study the efficiency of pre-trained LMs in application to the fold classification problem. To start with, we consider the state-of-the-art method DeepSF [9]. It solves the protein fold classification problem. By replacing its MSA-based feature-generation part with a pre-trained LM, we study how NLP approach affects learning the fold-related information. The whole framework is schematically represented in Figure 1. Further, we perform the same with other fold classification algorithms [7,14] as well as with end-to-end protein structure prediction methods [10,15].

2 Problem statement

2.1 General description

Proteins that share similarities in their structure are divided into classes – folds [12]. In this work we investigate a fold classification problem.

We use the following notation: Y is a set of all known classes, X is a set of all proteins with known folds and $f : X \to Y$ is a mapping between proteins and their classes.

Suppose, $x \notin X$ is a protein with unknown fold. We aim to build a model capable of classifying x as an element of one of the present classes, extending f on the set $X \cup \{x\}$.

2.2 Feature generation

In the fold classification problem x_i represents i-th protein description in the D-dimensional space. However, initially each protein is described by its amino acid sequence. Let Σ be a 20-letter alphabet of acid sequences.

We are given a set of $A = \{a_i \in \Sigma^+\}_{i=1}^n$. Let $g_{\theta,D} : \Sigma^+ \to \mathbb{R}^D$ be an embedding of the space of acid sequences into D-dimensional set of descriptors. These embeddings are parameterized by θ.

We separate A into training and validation set. Training set is used to formulate fold classification problem. Validation set is used to formulate feature generation problem.

$A = A_{train} \cup A_{val}$

Protein sequence descriptors are generated using $g_{\theta,D}$ as $x_i = g_{\theta,D}(a_i)$

2.3 Fold classification problem

Suppose there is a given set of pairs $S = \{(x_i, Y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}^D$ denotes the sequence descriptor of i-th protein and $Y_i \in \mathbb{Y} = \{(1,0,...,0),(0,1,0,...,0)....(0,...,0,1)\}$ denotes the i-th protein known class, $|\mathbb{Y}| = m$

We separate S into training and validation set. Training set is used to formulate fold classification problem.

$S = S_{train} \cup S_{val}$

Let $W = \{(w_1,...,w_m)|\forall i \in \{1...m\} w_i \in \mathbb{R}^D\}$ be a space of parameters for classification models. Let g_w be a model parameterized by w. We use e_k to denote a vector with 1 on k-th position, with zeros on all other positions.

$$e_k = \quad \begin{array}{ccc} (0,\ldots,0, & 1, & 0,\ldots,0) \\ & \uparrow & \\ & k & \end{array}$$

We work under the following assumption:

$P_w(Y = e_k|x) = \frac{exp(x^T w_k)}{\sum_j exp(x^t w_j)}$

Let $p_{w,x}(y)$ be discrete density of Y, $L_{Y,x}(w)$ be likelihood function of Y.

$p_{w,x}(y) = \prod_{d=1}^m (P_w(Y = e_d|x))^{y_d}$,

$L_{Y,x}(w) = \prod_{i=1}^n p_{w,x}(Y_i)$

Given S_{train}, classification problem is a maximization of likelihood on training set:

$L_{train}(w) = L_{Y_{train},x_{train}}(w) \to \max\limits_{w \in W}$

2.4 Feature generation problem

For fixed θ, D and descriptors x_i let's denote $\hat{w} = \arg\max\limits_{w \in W} L_{Y_{train},x_{train}}(w)$.

Feature generation problem is a maximization of likelihood function on validation set:

$L_{Y_{val},x_{val}}(\hat{w}) \to \max\limits_{\theta,D}$

2.5 Overview

The problem of determining proteins' class is separated into two parts: first, we encode amino acid sequences as D-dimensional descriptors, then these descriptors are used to train a prediction model.

3 Computational experiment

3.1 Data

We use SCOP2 [1,2] dataset. It contains information about proteins whose structure is already known. All the proteins in the database have classes assigned to them (folds). At present, there are 1501 known folds [2].

In SCOP2 proteins are represented by their amino acid sequences. Each sequence is stored as a string in the 20 letter alphabet and an ID. The length of the string varies from one protein to another. Search by ID allows database users to extract information about target proteins, including their fold classes.

3.2 Baseline solution

Our solution is a modification of the DeepSF [9] method. The designers of DeepSF use PSI-BLAST [3] to generate features from MSA as well as SCRATCH [11] to extract secondary structure (3 classes) and solvent accessibility (2 classes). In the core of this classification method PSI-BLAST is used during the first stage of prediction [11].

Hoe et al. then use these features as an input to the deep convolutional neural network. The depth of CNN used in DeepSF is 10. The output of the model is a vector of fold probabilities. Fold with the maximal probability is accepted as a target protein fold.

3.3 Proposed solution

DeepSF feature generation is entirely based on MSA. The main point of this work is to suggest another approach to feature generation which is independent of MSA.

We apply BERT to the language of amino acids. We take the representation of acid sequences BERT creates as a new set of features. Then we train DeepSF model on these features.

It shall be noted that we do not reject co-evolution information in our research. Instead of directly applying MSA we let BERT learn information about amino acid language. We suppose that the model learns evolution information implicitly.

3.4 Evaluation

We use standard cross-validation procedure to compute accuracy of our model. We then compare it to the accuracy score of DeepSF method based on MSA feature generation as well as several other state-of-the-art techniques.

4 Structure of the article

Divide your article into clearly defined and numbered sections and paragraphs.

4.1 Paragraph

Sections and paragraphs are numbered and have a brief heading.

5 Concluding Remarks

This section should provide the summary and explore the significance of the results achieved and list problems not yet solved. Results should be clear and concise.

References

[1] Antonina Andreeva, Dave Howorth, Cyrus Chothia, Eugene Kulesha, and Alexey G. Murzin. SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Research*, 42(D1):D310–D314, 11 2013.

[2] Antonina Andreeva, Eugene Kulesha, Julian Gough, and Alexey G Murzin. The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic Acids Research*, 48(D1):D376–D382, 11 2019.

[3] Medha Bhagwat and L. Aravind. Psi-blast tutorial. *Methods in molecular biology (Clifton, N.J.)*, 395:177–186, 2007. 17993673[pmid].

[4] Davide De Lucrezia, Debora Slanzi, Irene Poli, Fabio Polticelli, and Giovanni Minervini. Do natural proteins differ from random sequences polypeptides? natural vs. random proteins classification using an evolutionary neural network. *PLOS ONE*, 7(5):1–10, 05 2012.

[5] S. de Oliveira and C. Deane. Co-evolution techniques are reshaping the way we do structural bioinformatics. *F1000Res*, 6:1224, 2017.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[7] W. Elhefnawy, M. Li, J. Wang, and Y. Li. DeepFrag-k: a fragment-based deep learning approach for protein fold recognition. *BMC Bioinformatics*, 21(Suppl 6):203, Nov 2020.

[8] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv*, 2020.

[9] Jie Hou, Badri Adhikari, and Jianlin Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 12 2017.

[10] Shaun M Kandathil, Joe G Greener, Andy M Lau, and David T Jones. Deep learning-based prediction of protein structure using learned representations of multiple sequence alignments. *bioRxiv*, 2020.

[11] C. N. Magnan and P. Baldi. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, 30(18):2592–2597, Sep 2014.

[12] R. Dustin Schaeffer and Valerie Daggett. Protein folds and protein folding. *Protein engineering, design & selection : PEDS*, 24(1-2):11–19, Jan 2011. 21051320[pmid].

[13] Akif Uzman. Molecular biology of the cell (4th ed.): Alberts, b., johnson, a., lewis, j., raff, m., roberts, k., and walter, p. *Biochemistry and Molecular Biology Education*, 31(4):212–214, 2003.

[14] A. Villegas-Morcillo, A. M. Gomez, J. A. Morales Cordovilla, and V. E. Sanchez Calle. Protein fold recognition from sequences using convolutional and recurrent neural networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2020.

[15] Jinbo Xu, Matthew Mcpartlon, and Jin Li. Improved protein structure prediction by deep learning irrespective of co-evolution information. *bioRxiv*, 2020.

Received January 01, 2017