On amino acid vector embeddings, similarity scores, and protein subcellular localization

Aniket Murhekar

University of Illinois, Urbana-Champaign
aniket1602@gmail.com

Abstract. The unique sequence of amino acids that make up a protein impart to it distinct physical and chemical properties. Inspired by ideas in NLP like word2vec and sequence-based models, we create vector embeddings of amino acids which encode contextual information meaningful biochemical properties. We use these vector embeddings to compute substitution matrices for the problem of protein sequence alignment. We also use these embeddings together with sequence based models for the task of predicting protein subcellular localization.

1 Introduction

Proteins govern almost every major activity on the cellular level. The unique sequence of amino acids that make up a protein impart to it distinct biophysical and biochemical properties. A major challenge in biology is to learn precisely the function of a protein from its primary sequences of amino acids. Since advancements in full genome sequencing have provided us with a wealth of genome data, the possibility of applying computational tools to aid has opened up.

A well studied problem in bioinformatics is the prediction of protein subcellular localization [1, 10, 11]. Protein functions can vary depending on organelle they are located in. Knowledge of the subcellular localization of a protein can significantly improve target identification during the drug discovery process. For example, secreted proteins and plasma membrane proteins are easily accessible by drug molecules due to their localization in the extracellular space or on the cell surface. It is also known that aberrant subcellular localization of proteins has been observed in the cells of several diseases, such as cancer and Alzheimer's disease. However, experimentally determining the subcellular localization of a protein can be a laborious and time consuming task. Through the development of new approaches in computer science, coupled with an increased dataset of proteins of known localization, computational tools can now provide fast and accurate localization predictions [1].

Another common problem is finding alignments of protein sequences, which are used to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. These alignments are found by using a substitution matrix that describes the relative probability of one character in a sequence changing to another. In 1992, Henikoff

and Henikoff [2] developed a database of protein blocks based on sequences with shared motifs. Using this developed a matrix called BLOSUM (BLOck SUbstitution Matrix) to represent the similarity scores between amino acids. A problem in computational biology is to devise and evaluate different ways of coming up with these similarity scores, given that a lot more protein sequence data is now available.

To solve these problems, we borrow ideas from the field of natural language processing (NLP), where a breakthrough result was to represent words in high dimensional vector space, featurized so that words having similar meaning had similar vector representations. The main insight was that words having similar meaning tend to be observed in similar contexts (such as sentences). With this idea, the word2vec model [3] was created. Using word embeddings improved the performance on tasks such as sentence sentiment analysis and document classification. The analogue in the computational biology setting is to build vector representations of amino acids, with the idea that amino acids that occur in similar contexts a) impart similar functional properties to a protein containing the amino acid in the same context, and b) have a higher likelihood to replace one another. Thus we surmise that these amino acid vector embeddings encode some biochemical properties of amino acids and thus can improve the accuracy of protein function prediction or classification tasks, and provide meaningful substitution scores.

To summarize, we 1) explore and compare ways of building amino acids vector embeddings, 2) use them to calculate similarity scores and a substitution matrix, and compare it with BLOSUM, and 3) demonstrate the utility of these vector embeddings by using them in the classfication task of predicting subcellular localization.

2 Related Work

The idea of using vector embeddings for words [3] was first adapted to amino acids by Asgari et. al. [6]. They used their amino acid embeddings called ProtVec for protein sequence classification and to predict disordered proteins from structured proteins. Amino acid vector embeddings have also been used to generate similarity scores [7]. Longwell et. al. [9] generated the embeddings using 3D structural context of each residue and used them to classify amino acid mutations as neutral or destabilizing to T4 lyzosome, although they achieve a rather poor accuary. Bepler et.al. [8] trained bidirectional LSTMs to generate embeddings that incorporated structural information and used them on structural similarity tasks. Since sequence-based models are natural for the task of subcellular localization using primary amino acid sequence data, Armenteros et. al. [10] employed recurrent neural networks (RNNs) with long short-term memory cells (LSTMs). For the same task, Heinzinger et.al. [11] utilized deep bi-directional language models. Alternate ways of building and evaluating block substitution matrices have also been studied [4, 5].

3 Datasets

We used two datasets to generate amino acid embeddings. The Protein Data Bank (PDB) [12] is a collection of 493804 protein sequences, with an average sequence length of 253. SwissProt [13] is a collection of 560118 sequences, with an average sequences length of 360, which are reviewed and manually annotated. While these datasets have more than just sequence information such as structure factors, NMR restraints etc, we only relied on the primary sequence data.

We used the DeepLoc dataset [10] in the subcellular localization task. It is dataset of 140004 sequences of average sequence length 523, annotated with the subcellular localization. Ten classes are considered, which are: cell membrane, cytoplasm, endoplasmic reticulum, Golgi apparatus, lysosome/vacuole, mitochondrion, nucleus, peroxisome, plastid and extracellular.

4 Methods

4.1 Amino acid embeddings

Word2vec has two ways that generate dense vector representations of a symbol: continuous-bag-of-words (CBOW) and skip-gram (SG).

The two models have opposing tasks, CBOW attempts to predict a word given the context, while skip gram attempts to predict the context given a word. We briefly describe the CBOW model, while noting that the skip gram architecture is in essence a mirror image.

The CBOW model has three layers. The input layer consists of one-hot encoded input context words $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_C}\}$ for a word window of size C and vocabulary of size V. The hidden layer is an N-dimensional vector $\mathbf{h}$, which is connected to the input layer via a $V \times N$ weight matrix $\mathbf{W}$. The output layer is a one-hot encoded output word $\mathbf{y}$ in the training dataset. The hidden layer is connected to the output layer via a $N \times V$ weight matrix $\mathbf{W'}$.

The hidden layer is computed by simple averaging multipled by the weight matrix:

$$\mathbf{h} = \frac{1}{C}\mathbf{W} \cdot \sum_{i=1}^{C} \mathbf{x_i}$$

Next the inputs to each node in the output layer can be computed as $u_j = {v'}_j^T \cdot \mathbf{h}$, where v'_j is the j^{th} column of the output matrix $\mathbf{W'}$. Finally we compute the output of the output layer. The output is obtained by passing the input through the soft-max function,

$$y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u'_j)}$$

The weights $\mathbf{W}$ and $\mathbf{W'}$ are learnt through backpropagation using stochastic gradient descent. Since the objective is to maximize the conditional probability of the output word given the context, the loss function is its negative log likelihood.

4.2 Similarity scores

Having generated these vector embeddings, we calculated similarity scores between amino acids as the cosine of their vector representations:

$$\text{sim}(a_i, a_j) = \mathbf{v_i} \cdot \mathbf{v_j}$$

where a_i and a_j are two amino acids with vector embeddings as $\mathbf{v_i}$ and $\mathbf{v_j}$. Thus a substitution matrix S whose entries are given by $S_{ij} = \text{sim}(a_i, a_j)$ can be computed.

To compare against BLOSUM, we evaluated the relative entropy (also known as KL divergence) of the matrix S. Relative entropy is 0 when the target (or observed) distribution of pair frequencies is the same as the background (or expected) distribution and increases as these two distributions become more distinguishable. We calculated the background distribution p_i of amino acids using the UniProt dataset. If q_{ij} is the pairwise replacement provided by the vector embeddings, we have that $s_{ij} = \log_2 \frac{q_{ij}}{p_i p_j}$. The relative entropy H is calculated as:

$$H = \sum_{i,j} q_{i,j} s_{i,j}$$

4.3 Subcellular localization task

Feature selection For the task of subcellular localization, it is known that most of the information is known to reside in the beginning (N-terminus) and end (C-terminus) of the sequence. Since the average sequence length is around 500 we consider the only the first and last 50 amino acids in the protein to build our feature vector.

Model We built a simple neural network classifier for the subcellular localization task. The network had three layers, the first being an embedding layer, for which we used the pretrained amino acid embeddings. The second layer was a Long Short-Term Memory (LSTM) layer with 32 units. Finally we had a dense, fully connected layer with softmax activation which output vectors in dimension 10, the number of classes. We used categorical cross entropy as our loss function.

5 Experiments

Since the number of amino acids are 20, we considered vector representations of dimension 15-20. We observed that lower dimensions provided slightly better accuracy in the subcellular localization task than higher dimensions. We also observed that accuracy slightly improved by changing the embedding creating method from skip-gram to CBOW. Out of the 14004 sequences in the DeepLoc dataset, 10107 sequences were used for training, 1124 for validation and 2773 for testing. We also experimented by changing the second layer in the neural

network model from LSTM to a Gated Recurent Unit (GRU) layer, but did not observe any significant change in the accuracy. The model to train the word embeddings took about thirty minutes, whereas training the neural network for the subcellular localization task took around two hours for 128 epochs. Reducing the feature size significantly reduced the time required.

The code along with instructions on how to run and links to download the data is provided at `https://github.com/murhekar/aa-vec-embeddings`.

6 Results and Discussion

To test if the amino acid vector embeddings encoded meaningful biochemical information, we employed dimensionality reduction methods like Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) to project these vectors down to two dimensions. We noticed that amino acid sharing similar biochemical properties such as polarity, aromaticity and hydrophilicity were clustered together, thus validating our hypothesis.

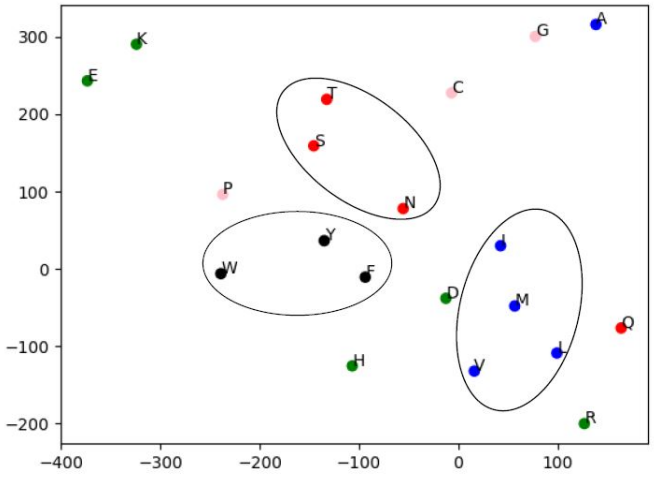

Fig. 1. Amino acid embeddings projected down to two dimensions using t-SNE. Threonine(T), Serine(S), and Asparagine(N) have polar uncharged side-chains. Tryptophan(W), Tyrosine(Y) and Phenylalanine(F) have aromatic and hydrophobic side chains. Isoleucine(I), Leucine(L), Methionine(M) and Valine(V) have aliphatic hydrophobic side chains.

We observed that relative entropy was found to be comparable to BLOSUM when the embeddings were generated using CBOW as compared to skip-gram, thus suggesting that the distribution provided by the CBOW is closer to the

background distrubution than the skip-gram model. This can be explained using the fact that the CBOW training task of predicting amino acid probability from context is more likely to capture more contextual information relevant to generating similarity scores.

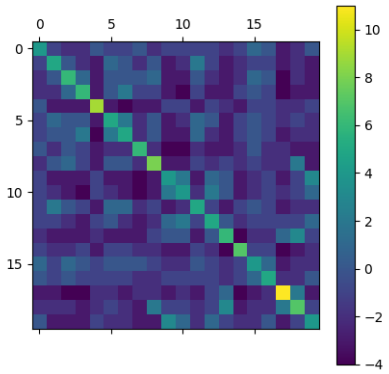

Fig. 2. BLOSUM62

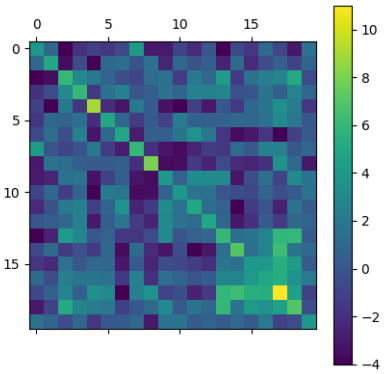

Fig. 3. New substitution matrix

Table 1. Relative entropy of the substitution matrices.

	BLOSUM	CBOW	Skip-gram
Rel. Entropy	84.8	89.4	96.7

We observed better accuracy when the embeddings were generated using CBOW as opposed to skip-gram. Despite the accuracy of the model being only around 50-65%, we note that the word embeddings perform significantly better than when the amino acids are encoded as one-hot vectors or when the encoding of an amino acid is the corresponding row of the BLOSUM matrix. We also note that using pretrained amino acid word embeddings resulted in a slightly better accuracy than training the embeddings along with the network for the classification task. We conjecture that on performing more rigorous hyperparameter tuning and more epochs for training the accuracy can be improved, even for this simple architecture. Another possible reason could be that the embedding vectors were trained on datasets (Uniprot and PDB) that had shorter sequences on average as compared to the DeepLoc dataset used for the classfication task. Once again we observe that the CBOW model performed better as compared to the skip-gram model of generating embeddings. Changing the second layer from

LSTM to GRU in the subcellular localization neural network did not significantly alter the accuracy.

Table 2. Accuracy of embedding models employed in the subcellular localization task.

Embedding model	One-hot	BLOSUM	Skip-gram	CBOW
Train	31.77	31.03	53.48	65.91
Test	28.92	31.66	44.21	49.84

7 Future work

Future directions to take this work forward include experimenting with different architectures to generate the vector embeddings like bidirectional LSTMs and attentional encoders [8,10,11]. Including more data like 3D protein structure can generate embeddings that encode more contextual information, thus increasing their utility for other protein function prediction or classification tasks [9]. The substitution matrices produced by these embeddings can be evaluated and compared with the BLOSUM matrix on sequence alignment tasks and measuring the coverage [4,5]. For the specific task of subcellular localization, the model can be improved using different architectures, better feature selection and hyperparameter tuning. Finally, we can use these amino acid embeddings on different protein function prediction or classification problems.

8 Conclusion

Amino acid vector embeddings are a way to encode meaningful biochemical information by representing amino acids as vectors. We find that amino acid vector embeddings can give rise to different methods of calculating amino acid similarity scores. We also found them to be useful in protein function prediction and classification tasks.

References

1. Protein subcellular localization prediction, `https://en.wikipedia.org/wiki/Protein_subcellular_localization_prediction`.
2. Amino acid substitution matrices from protein blocks. Henikoff and Henikoff, Pubmed 1992.
3. Distributed Representations of Words and Phrases and their Compositionality. Mikolov et. al., NIPS 2013.
4. Addressing inaccuracies in BLOSUM computation improves homology search performance. M. Hess et. al., BMC Bioinformatics, 2016.
5. Selecting the Right Similarity-Scoring Matrix. WR Pearson, Curr. Prot. in Bioinformatics, 2018.

6. ProtVec: A Continuous Distributed Representation of Biological Sequences. e. Asgari et. al., PLOS ONE, 2015.
7. Graph matching, pattern learning and protein modelling. W. Qian, Senior Thesis, 2017.
8. Learning protein sequence embeddings using information from structure. T. Bepler et. al., arXiv preprint, 2019.
9. Res2Vec: Amino acid vector embeddings from 3d-protein. Longwell et. al., 2019.
10. DeepLoc: prediction of protein subcellular localization using deep learning. Armenteros et. al., PubMed, 2017.
11. Modeling the Language of Life Deep Learning Protein Sequences. Heinzinger et. al., biorXiv preprint, 2019.
12. The Protein Data Bank. H.M. Berman et. al., Nucleic Acids Research, 2000 `http://www.rcsb.org/`
13. UniProt: a worldwide hub of protein knowledge. The UniProt Consortium, Nucleic Acids Research, 2019. `https://www.uniprot.org/`