Electronics and Computer Science
Faculty of Physical and Applied Sciences
University of Southampton

Brian Formento

27 May 2018

Secondary protein structure prediction

Comp3212: Computational Biology

1 Background

Globular proteins have three main structures (primary, secondary and thirdly). The primary, being the sequence of amino acids and the lowest level of complexity forms the secondary structure, or how a local polypeptide chain is shaped, the shape comes from different areas acquiring different forms, the main forms are alpha-helix (H), beta-sheet (E) or coil (C) which encompasses every other one. Lastly, the tertiary structure is used to describe the overall 3D structure of the polypeptide chain.

Due to the expensive procedure, such as X-ray crystallography, used to detect secondary structures, it is sometimes unfeasible to carry out such measurements. Therefore, a cheaper alternative is using cheaper to obtain primary structure data, and through the use of algorithms predict the look of the structure.

In recent years the development of machine learning methods has seen a rapid increase in innovation in the field. Most recently networks capable to learn time series features have been developed. Therefore this assignment proposes the use of an LSTM network for secondary protein structure prediction as, although the available software does use machine learning, there haven't been found any LSTM based services.

Two datasets have been found and analysed. Firstly the recommended UCI (2) dataset contained 91 amino acid sequences and has been used for the Terrence Sejnowski experiment using a neural network, in 1988. The dataset has a bad layout, each line containing the amino acid and its respective fold type. This is quite obsolete, but it's not surprising as the research paper was released before the invention of python. The dataset contains all 20 amino acids, the folds are divided into 3 types, ‘E’ alpha helix, ‘H’ beta-sheet and ‘-’ being everything else.

An alternative dataset is the TS1199, composing of over 2000 training samples and 1199 test samples. This dataset is in FASTA format, which means is very easy to use with python. (9)

2 Tested algorithms

Two algorithms have been tested

2.1 Chou fastman

In 1978 the Chou Fastman method was one of the first developed algorithms for secondary protein structure prediction. The team used X-ray crystallography techniques to detect the frequency at which amino acid bases occur in certain secondary structure regions, by doing so they developed a value matrix, where every amino acid had a probability value associated with it. The algorithm then uses probabilistic methods to categorise an amino acid, classifying it with its corresponding secondary structure. Chou and Fastman declared the accuracy to be 70%, although other sources put this figure closer to 50% to 60% this has been accredited to Chou originally testing on the training dataset and because of the empirical nature of this algorithm (7)(8), this especially lower compared to more modern machine learning techniques. In fact, an analysis was carried out on the UCI dataset, it was composed of 21.8% alpha helix, 35.7% beta sheet and 42.5% coil, the coil percentage is very close to the advertised 50%.

2.1.2 Testing

This algorithm was tested using a forked implementation (5) giving mixed results while using the UCI dataset, which holds 91 sequences. A mean of every sequence accuracy was taken and overall this was 46.8%, much lower than the advertised 50%. This can, however, be accredited to some anomalies in the results, where some were as low as 10%.

2.2 Jpred

Due to the bad results obtained from Chou Fasman a different and more modern algorithm (Jnet) was tested. Jnet can be found on (Jpred)(6)

This algorithm uses 2 neural networks developed with the SNNS library from Stuttgart University. The first one uses a 17 sliding window input, 9 hidden layers which get compressed to 3 output nodes. These outputs are then fed into the second network using a 19 window. The second network also has 9 hidden layers and 3 output nodes. It was originally trained on a modified version of the cb513 dataset, where strands of less than 30 amino acids were removed and the remainder passed through the PSI-BLAST alignment algorithm (10). This has been made easily accessible through an aesthetically pleasing UI interface.

2.2.2 Testing

An analysis was carried out on the output of a batch of 10 sequences gathered from the (9) dataset, for each sequence the prediction accuracy was calculated and a mean standing at 78% generated (using python). This is much lower than the advertises 82%. (10)

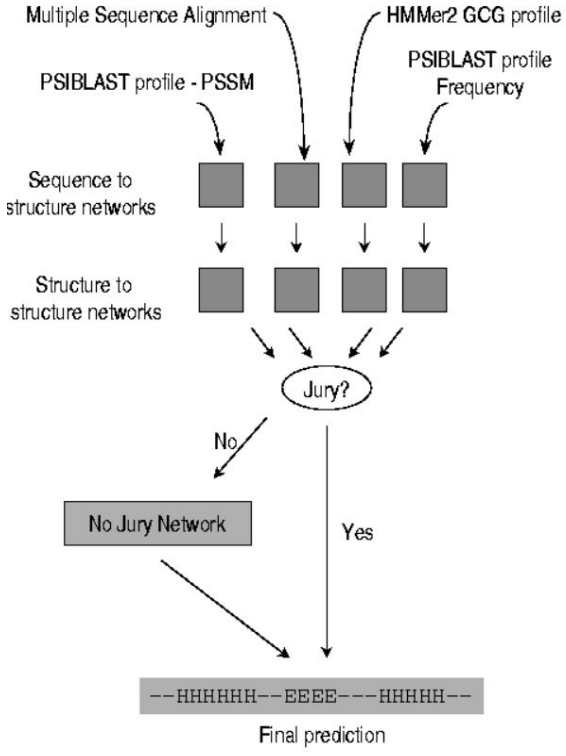


Fig 1: The prediction validation process on Jpred uses a 'jury system' consisting of other NN trained on alignment data where if all network outputted identical prediction the step was valid, otherwise the

‘no jury’ prediction was used to train a different network. Where those ‘no jury’ predictions were consequently replaced with predictions from this new network. (10)

3 Custom algorithm

As mentioned in the background an LSTM network has been built. This LSMT network was built using the Keras deep learning library with a Tensorflow engine running in the background.

3.1 LSTM

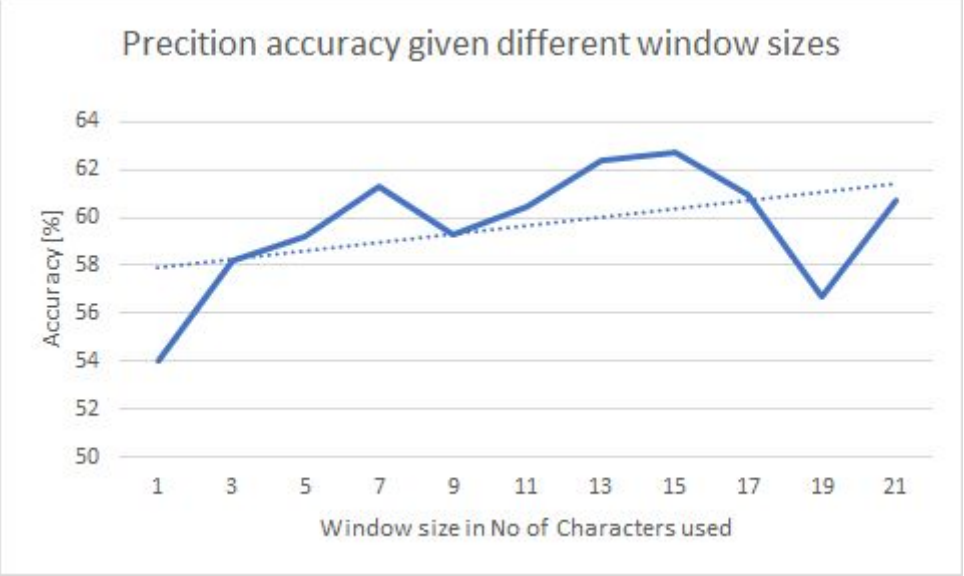
The model’s architecture is as it follows:

- LSTM layer 7 by 20 input (hot encoded), outputting a sequence of 126 length vectors.
- 2 more LSTM layers outputting a sequence of 64 length vectors in series.
- 1 last LSTM layer outputting a single 32 length vector.
- 20% dropout.
- A dense layer with 3 outputs (hot encoded) with a softmax activation function, to be used with a categorical classification method.

The datasets to train and test this model were both UCI (2) and TS1199 (9), however (9) had to be reduced by one third due to python’s memory limitations, this could be improved by using a Keras feature called generators, which allows loading the data, train the model and then clear the memory dynamically from a file or folder. Not putting any strain on the environment’s resources.

The data was initially preprocessed, as both datasets are in different formats. Therefore two functions allow accessing both datasets, creating a consistent output format.

The first test using UCI (2) gave 62.7% accuracy while using TS1199 (9) gave 63.4%. This is not surprising, as TS1199 (9) allowed to train on more data. Since this test was carried out with a window of 15, smaller than the 17 window used in (10) a further test to see how accuracy changes while varying window size was carried out, using the UCI (2) dataset.



Graph 1: The graph shows an upward trend, the larger the window size the larger it an accuracy is to be expected. It is clear thou that this trend starts falling after 17, where the optimal is 15.

This algorithm's accuracy is above 41% and therefore not random. Furthermore compared to the Chou Fasman method it is more accurate. However, this would be an unfair comparison as deep learning has been used. Therefore this accuracy should be compared to the T. Sejnowski method released in 1988 (2) which achieved a 64.3% accuracy. My algorithm is therefore 1.6% worst on the same dataset, while using, in theory, a more powerful and modern technique.

Comparing to Jpred would also be unfair as parsing the initial amino acid sequence through PSI-BLAST clearly improved the overall accuracy, as shown by their published paper.

Overall a good result has been achieved, which gives confidence that fine tuning this base model could achieve above 64.3%, There is to say that to further improve this accuracy it would be most beneficial to do a test using inputs from PSI-BLAST, as using this method showed promising results with Jnet.

References

(1) Washington.edu, avaiable at https://courses.washington.edu/conj/protein/protein.htm

(2) Ning Qian and Terrnece J. Sejnowski (1988), "Predicting the Secondary Structure of Globular Proteins Using Neural Network Models" in Journal of Molecular Biology 202, 865-884. Academic Press, avaiable at http://users.ecs.soton.ac.uk/mn/ArtificialIntelligence/QianSejnowskiPaper.pdf

(4) Chou PY, Fasman GD (1978). "Prediction of the secondary structure of proteins from their amino acid sequence". *Adv Enzymol Relat Areas Mol Biol*. **47**: 45–148.

(5) Samuel A. Rebelsky, Gridell, avaiable at https://github.com/ravihansa3000/ChouFasman/blob/master/ChouFasman.py

(6) Drozdetskiy A, Cole C, Procter J & Barton GJ. Nucl. Acids Res.
(first published online April 16, 2015, available at http://www.compbio.dundee.ac.uk/jpred/

(7) Nishikawa K. Assessment of secondary-structure prediction of proteins comparison of computerized Chou-Fasman method with others. Biochim Biophys Acta. 1983;748:285–299.

(8) Hang Chen Fei Gu and Zhengge Huang, 2006, Improved Chou-Fasman method for protein secondary structure prediction, BMC Bioinformatics

(9) available at http://sparks-lab.org/server/SPIDER2/
(10) James A. Cuff and G. J. Barton, Application of multiple sequence alignment profiles to improve protein secondary structure prediction. Laboratory of Molecular Biophysics.
(11) Ning Qian and Terrnece J. Sejnowski (1988), "Predicting the Secondary Structure of Globular Proteins Using Neural Network Models" in Journal of Molecular Biology 202, 865-884. Academic Press