

SALIENCY MAP ON CNNs FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Guillermo Romero Moreno

Mahesan Niranjan

Adam Prugel-Bennett

School of Electronics and Computer Science

University of Southampton, Southampton, UK

Email: grm1g17@soton.ac.uk, m.niranjan@southampton.ac.uk, apb@ecs.soton.ac.uk

ABSTRACT

Deep learning has been successfully transferred to the biomedical field, achieving higher performance but bringing opaqueness. While interpretability techniques are under current development, only a few authors have brought them to the biomedical field. This work aims to apply one of such techniques —saliency maps— in the context of protein secondary structure prediction. For doing so, a convolutional neural network was first trained, saliency maps obtained from it, and different ways of aggregating them have been developed to gather meaningful insights on the network’s learning of the underlying problem structure. The analysis of saliency maps led to interesting findings, namely the irrelevance of one-hot amino-acid inputs and the higher importance of right positions for the prediction of α -helices. These techniques can be of double value: they help biologists to get a better understanding of the underlying processes and machine learning researchers can spot previously uncovered flaws.

Index Terms— Saliency Maps, Convolutional Neural Networks, Protein Secondary Structure Prediction, Interpretability.

1. INTRODUCTION

Secondary structure prediction is a long-time studied problem in bioinformatics. The 3D structure of a protein determines the function it is going to adopt in the cell. However, the protein structure cannot be easily measured and any attempt to do so remains costly. A more feasible alternative is utilising computational tools to make the predictions having as a base the amino-acid sequence (easy to obtain through DNA sequencing) and the proteins whose structure is already known. Predicting the secondary structure is often regarded as a middle step for tackling the much harder problem of predicting the 3D structure. The protein secondary structure prediction problem is a sequence structural tagging problem: each element (amino-acid) of the protein sequence has to be assigned a class (secondary structure). There are 21 different types of amino-acids and eight possible goal classes, which are composed of 3 types of helices (H, G, I), two types of β -sheets (B, E), and three types of coils (T, S, L) [1]. Other relevant

features of the amino-acids can also be added as inputs to facilitate the classification process. A common one whose addition brought a significant performance improvement is the Position Specific Substitution Matrices (PSSM) [2], which encode the evolutionary probability of finding substitutions in each element of the amino-acid chain. The final input sequence would have length l (variable from protein to protein) and width 42: the one-hot encoded amino-acid plus the 21 extra values of the PSSM, normalized to a range between zero and one [3]. While at the early stages the secondary structure prediction problem was mainly tackled with statistics, towards the end of the century the application of neural networks became prevalent [4]. A new generation of deep learning approaches started recently with [5], who implemented a Generative Stochastic Network fed by a 1D Convolutional Neural Network (CNN) architecture, and later works that already included 1D CNNs with five or more layers [6, 7] or recurrent neural networks [8, 9], which are deep in the sense of signals being processed for many time-steps.

Saliency maps are a visualisation technique for sensitivity analysis. A saliency map has the same dimensions as the input and contains their importance values, e.g. their contribution to the output. Depending on their calculation method, saliency maps can be broadly grouped into *perturbation-based approaches* (make modifications on the input and assess changes in the output) or *backpropagation-based approaches* (utilise the gradient of the output respect to the input for obtaining the importance information) [11]. Although the first group is intuitive and useful for small input spaces, it becomes quickly intractable when the input size grows, as all possible combinations of inputs should be examined for a complete analysis. The second group —especially designed for neural networks— allows the computation of importance scores in a single backward pass, so its computational complexity significantly improves and hence it is the preferred option for such architectures.

Back-propagation approaches can be thought as a linear approximation of the classification function around a sample input point x_0 by applying a first-order Taylor expansion, as

introduced in [12]:

$$f(x) \approx w^T x + b, \quad (1)$$

$$w = \left. \frac{\partial f}{\partial x} \right|_{x_0}. \quad (2)$$

In their simplest form, the saliency maps of back-propagation methods are equivalent to the gradient value on the input [12]. A second approach [13] would multiply the gradient by the input values to leverage out the gradients that don't carry relevant information. A last wave of methods proposes including a reference point and hence more closely resembling the Taylor approximation. The main examples of this trend are integrated gradients [14], deep Taylor decomposition [15] and DeepLIFT [11]. They overcome problems of previous methods such as saturation or discontinuities in the gradient, although they bring the extra difficulty of choosing an appropriate reference point.

2. PREVIOUS WORK

Perturbation-based approaches are prevalent, with perturbation as genetic mutations [16, 18], small sliding windows with random genetic code [19] or known motifs [20], among others. Gradient-based approaches have barely been translated to the biological field. Lanchantin et al. [21] include saliency maps with the form of gradient \times input for TF binding site classification. They extracted the window with the highest score from each saliency map and compared them with a database of known motifs, matching almost half of the motifs thus produced. Shrikumar et al. [11] developed the reference-based saliency map technique DeepLIFT and simulated a motif detection task within a genomic sequence to prove its effectiveness. Finnegan and Song [22] utilised Markov chain Monte Carlo methods to withdraw samples from the maximum entropy distribution around a single sequence and assessed the importance scores by looking at the variance of the samples at each position. This method was applied to a previously trained DNA-protein binding CNN and proved to have better results than DeepLIFT.

All these methods address classification problems where there is a single output (classification task) for each sequence. A significant difference between this work and previous papers that make use of saliency maps is that they focus on many-to-one classification problems (one output class per input sequence/image), whereas our classification task is many-to-many (each position of the sequences is assigned a class), producing as many saliency maps as positions in a sequence. To the best of our knowledge, interpretability techniques have not been applied yet to this sort of problems.

3. METHODS

The experiments used the database produced and made public by Zhou and Troyanskaya [5]. It includes two sub-sets

Q8 grouping		Explanation	%
α -helix	H	Helix with 4 turns	34.54
3_{10} -helix	G	Smaller helix with 3 turns	3.91
π -helix	I	Bigger helix with 5 turns	0.02
β -bridge	B	Isolated β -bridge	1.03
β -strand	E	Participates in β -ladders	21.78
Turn	T	Turns smaller than a helix	11.28
Bend	S	Curved piece	8.26
Loop	L	Sometimes also as coil (C)	19.19

Table 1. Targets for the secondary structure prediction problem, as defined by Kabsch and Sander [1] in their Dictionary of Secondary Structure of Proteins (DSSP) and their presence on the training set.

(training and test, with 5534 and 514 protein sequences of varying length, respectively) of proteins that come from different sources after removing the proteins that share 25% or more similarity, thus ensuring that the test set is composed of totally new samples. The proteins in the dataset already come in one-hot form, along with their Q8 class in one-hot as well and PSSM values. The dataset is heavily imbalanced, as it can be seen in Table 1

The network architecture is composed of three successive convolutional neural networks and a dense layer on top. Each of the convolutional layers contains three sets of filters of size 3, 5 and 7, respectively, with 16 filters per size. There are skip connections at every convolutional layer. The dense layer has 200 neurons and is connected to the softmax output layer. The convolution operations are carried out with padding at each end of the sequence to preserve the length throughout the process. The total window size of the network is 19, meaning that for making a single secondary structure classification the network obtains information from 9 adjacent positions at each side. The network has been built and trained using the open-source code developed by Jurtz et al. [9].

Saliency maps are calculated by the conventional technique of computing the gradient of the output with respect to the inputs and multiplying it by the value of the input (gradient \times input) [23]. Every single position in a sequence produces a saliency map that spans the width of the input vector of size 42 and 9 positions to each side, due to the architecture's window size of 19. Each output class has its independent saliency values, so the total size of a position saliency map is $8 \times 42 \times 19$.

The presence of overlapping saliency maps allows for different ways in which to aggregate them to extract meaningful information. If we focus on a sequence of length l and want to obtain a single sequence-specific saliency map, we can add up the overlapping areas to form a saliency map of size $8 \times 42 \times l$. By changing the focus to a broader look on what the network has learnt, the addition of the saliency maps for the positions in all sequences could create a single saliency map of size $8 \times 42 \times 19$ that shows an average behaviour of the network.

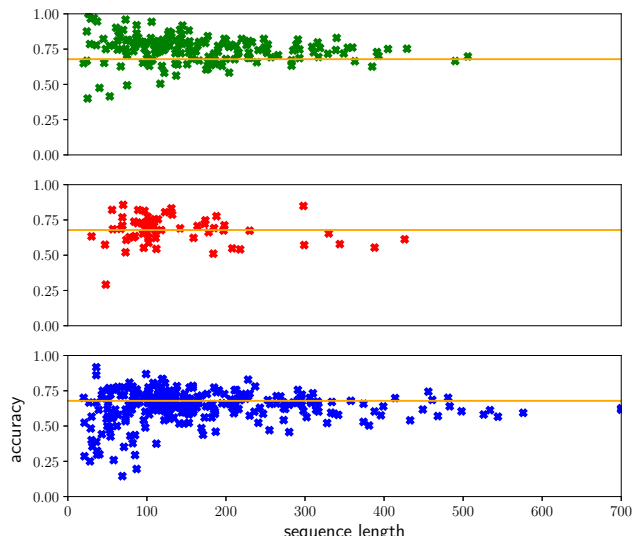


Fig. 1. Mean sequence accuracy against sequence length for the proteins in test set. Each point represents a single protein sequence and the horizontal line is the total mean accuracy. On top, sequences with majority of helices (H, G, I); β -sheets (E, B) at the middle; and coils (L, S, T) below. Best accuracies are achieved with high percentage of helices and worst accuracies with majority of coils.

From this map, we can extract general information about a particular class (creating a class-specific saliency map) or about a particular input (PSSM-specific saliency map).

4. RESULTS

The network described in the previous section has been trained for 400 epochs with regularisation parameter $\lambda = 10^{-4}$, and learning rate $\mu = 10^{-4}$. Five of such networks with different weight initialisation are trained, and the set of weights for each of them with highest validation accuracy¹ is taken to form an ensemble that reaches an accuracy of 69.23% on the test set, not far from the 71% reached by the state-of-the-art [3]. The aim of this work is not to outperform the state-of-the-art predictions, but to build a network with a moderately simple structure (to keep the calculation times of saliency maps on reasonable levels) and fair performance. We believe that the techniques of analysis here presented and the conclusions withdrawn from them can be transferred to current state-of-the-art methods without losing validity.

Figures 1 and 2 show more information on the network performance. Figure 1 shows the distribution of per-sequence accuracies over different sizes. As it could be expected, the variance in mean accuracy increases with shorter sequences. Higher accuracies can be expected from sequences rich in helices and lower accuracies for sequences high in coils. Figure

true	L	11036	38	3072	271	0	945	1209	1349
	B	515	47	325	13	0	84	92	105
	E	2171	30	14427	60	0	582	331	415
	G	541	5	217	906	0	766	114	583
	I	5	0	2	0	0	21	0	2
	H	748	7	459	320	0	23743	156	724
	S	2898	13	942	162	0	662	2063	1576
	T	1568	3	538	463	0	1631	611	5199
		L	B	E	G	I	H	S	T
predicted									

Fig. 2. Confusion matrix of the test set. Classes I and B are scarce in the dataset; classes G, H and T have relatively high levels of confusion because they are all composed of turns, but with different lengths; loops (L) are a loose category and their confusion is also high.

2 displays the resulting confusion matrix of the predictions on the test set. It is similar to the ones obtained on other state-of-the-art networks [6]. It is noticeable that the π -helix (I) has never been given as prediction; its presence is so rare in the dataset that the machine keeps a higher accuracy by ignoring it. β -bridges (B) are also largely misrepresented for the same reasons. Loops present high levels of confusion with other classes, probably due to the arbitrary discretisation into eight classes, while it has been pointed that it is not that clear the class assignment at the transitions between structures and coils [24].

Saliency maps can be used as additional evidence for theories around protein secondary structure. For instance, one point of concern has been the inclusion of inputs with different nature: the one-hot amino-acid along the PSSM dense vector. Some authors [8, 7] embedded the one-hot vector into a denser space, reporting a small marginal improvement in accuracy (0.5% and 0.4%, respectively). Spencer et al. [25] reported a 2% Q3 improvement by not including the one-hot amino-acids at all. Saliency maps provide information of about the importance that different parts of the input have, so direct comparison of both kinds of inputs can be made by looking at their associated saliency values. To do so, each saliency map is split into two halves, corresponding to amino-acid and PSSM, respectively, and all the values of each half are summed up in absolute value to form a single saliency score. The comparison of such scores for all positions in the dataset is made in Figure 3, revealing that in the great majority of positions the PSSM inputs had four times or more

¹On a validation subset of the training set, with 512 sequences.

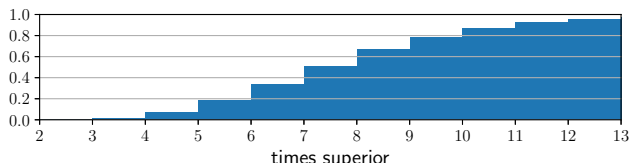


Fig. 3. Cumulative histogram of how much bigger are PSSM saliency scores as compared to amino-acid saliency scores.

Labels	Architecture	Accuracy
Q8	Original	69.23%
	PSSM-only	69.36%
H / non-H	Right positions only	86.60%
	Left positions only	80.81%

Table 2. Accuracies on the test set for different network configurations. Ensembles of networks performed better by ignoring one-hot amino-acids. Networks that look on a window of size nine at the right side predict class H better than when they regard the nine left positions.

relevance for making the classification decision, with around half of them having seven times or more relevance. We further validate these findings by training a second network that uses PSSM inputs but ignores the one-hot amino-acids, all the other things remaining equal. An ensemble of five of these networks reaches an accuracy of 69.36% on the test set, 0.14% points higher than the original configuration (see Table 2).

Another useful application of saliency maps can be the study of spatial importance around the predicted position. Figure 4 includes the average saliency spatial profiles for each class. The construction of each profile has been made as follows. For each sequence position belonging to the class that was predicted right, the 42x19 slice of the saliency map corresponding to the class is extracted and summed up over the feature dimensions, leading to a profile vector of size 19. The figure shows the average of all of such profiles for each class. Sequences coming from both the training and test set were used for this result. The profiles reveal what the spatial influence for each class. For instance, the α -helices (H) hold some periodicity at the right side and large asymmetry. The periodicity could go in line with the structure of helices, which are a succession of turns. The asymmetry can point to a strong dependency on posterior amino-acids when the protein chain is built instead of what was coming before. To validate this finding, we re-train two new networks for binary classification of H / non-H. The networks have an identical structure to the previous ones with the only modification being that one of them ignores the input from the left positions and the other the ones from the right positions². While the

²This is achieved by zeroing out different halves of the parameters on the convolutional filters.

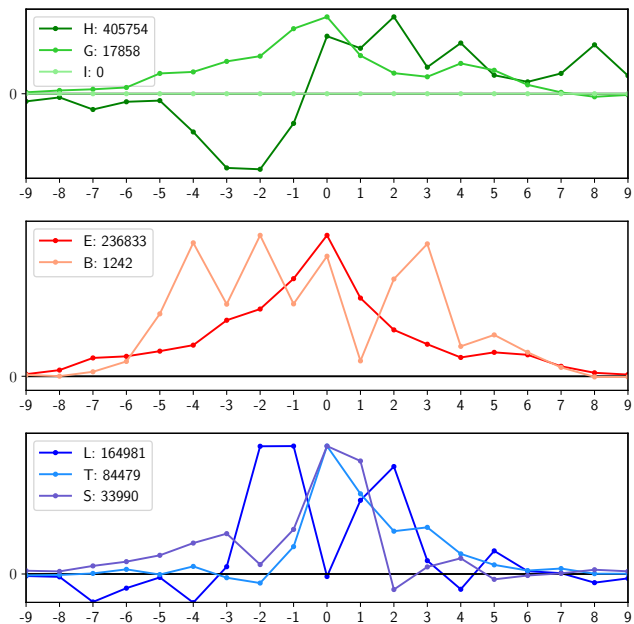


Fig. 4. Average saliency profiles for the eight classes. On top, the three helix classes; in the middle, the two β classes; below, the three coil classes. The legends show the number of profiles averaged over for each class. We want to focus on the distribution of each class over the window positions, so each line has been independently normalized for a clearer visualisation of each shape and scales only shared the zero reference.

network making use of the left positions obtains an accuracy of 80.81% on the test set, the one utilising the right positions achieves 86.60%, supporting the idea that future positions are more significant in the formation of an α -helix (Table 2).

5. CONCLUSIONS

Sensitive analysis help with dealing with the opaqueness of deep neural networks. Saliency maps are one of such techniques which have been extensively applied in the computer vision field but has barely been transferred to biomedical research. In this work, we have successfully applied saliency maps to the secondary structure prediction problem and shown that they can reveal insights from what the network has learnt, such as the irrelevance of one-hot amino-acid inputs or the strong dependence on future positions for the formation of α -helices.

6. REFERENCES

- [1] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983.
- [2] Y. Yang, J. Gao, J. Wang, R. Heffernan, J. Hanson, K. Paliwal, and Y. Zhou, "Sixty-five years of the long march in protein secondary structure prediction: The final stretch?," *Briefings in Bioinformatics*, 2018.
- [3] A. Busia and N. Jaitly, "Next-Step Conditioned Deep Convolutional Neural Networks Improve Protein Secondary Structure Prediction," *arXiv:1702.03865v1*, 2017.
- [4] B. Rost and C. Sander, "Improved prediction of protein secondary structure by use of sequence profiles and neural networks.," *Proceedings of the National Academy of Sciences of the United States of America*, 1993.
- [5] J. Zhou and O. G. Troyanskaya, "Deep Supervised and Convolutional Generative Stochastic Network for Protein Secondary Structure Prediction," 2014.
- [6] C. Fang, Y. Shang, and D. Xu, "MUFold-SS: Protein Secondary Structure Prediction Using Deep Inception-Inside-Inception Networks," sep 2017.
- [7] J. Zhou, H. Wang, Z. Zhao, R. Xu, and Q. Lu, "CNNH-PSS: Protein 8-class secondary structure prediction by convolutional neural network with highway," *BMC Bioinformatics*, 2018.
- [8] Z. Li and Y. Yu, "Protein Secondary Structure Prediction Using Cascaded Convolutional and Recurrent Neural Networks," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.
- [9] V. I. Jurtz, A. R. Johansen, M. Nielsen, J. J. Almagro Armenteros, H. Nielsen, C. K. Sønderby, O. Winther, and S. K. Sønderby, "An introduction to deep learning on biological sequence data: Examples and solutions," *Bioinformatics*, vol. 33, no. 22, pp. 3685–3690, 2017.
- [10] C. Olah, A. Mordvintsev, and L. Schubert, "Feature Visualization," *Distill*, 2017.
- [11] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning Important Features Through Propagating Activation Differences," apr 2017.
- [12] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv.org*, 2014.
- [13] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, 2015.
- [14] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," 2017.
- [15] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K. R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognition*, 2017.
- [16] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nature Biotechnology*, vol. 33, no. 8, pp. 831–838, 2015.
- [17] D. Quang and X. Xie, "DanQ: A hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences," *Nucleic Acids Research*, 2016.
- [18] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning-based sequence model," *Nature Methods*, vol. 12, no. 10, pp. 931–934, 2015.
- [19] R. K. Umarov and V. V. Solovyev, "Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks," *PLoS ONE*, vol. 12, no. 2, 2017.
- [20] D. R. Kelley, J. Snoek, and J. L. Rinn, "Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks," *Genome Research*, 2016.
- [21] J. Lanchantin, R. Singh, B. Wang, and Y. Qi, "Deep Motif Dashboard: Visualizing and Understanding Genomic Sequences Using Deep Neural Networks," 2016.
- [22] A. Finnegan and J. S. Song, "Maximum entropy methods for extracting the learned features of deep neural networks," *PLoS Computational Biology*, 2017.
- [23] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not Just a Black Box: Learning Important Features Through Propagating Activation Differences," *arXiv:1605.01713 [cs]*, 2016.
- [24] B. Rost, "Review: Protein secondary structure prediction continues to rise," 2001.
- [25] M. Spencer, J. Eickholt, and J. Cheng, "A Deep Learning Network Approach to ab initio Protein Secondary Structure Prediction," *Ieee/Acm Transactions on Computational Biology and Bioinformatics*, 2015.