Noname manuscript No.
(will be inserted by the editor)

PaleAle 5.0: Prediction of protein relative solvent accessibility by Deep Learning

Manaz Kaleel · Mirko Torrisi · Catherine Mooney and Gianluca Pollastri

Received: date / Accepted: date

Abstract Predicting the three-dimensional structure of proteins is a long-standing challenge of computational biology, as the structure (or lack of a rigid structure) is well known to determine a protein's function. Predicting relative solvent accessibility (RSA) of amino acids within a protein is a significant step towards resolving the protein structure prediction challenge especially in cases in which structural information about a protein is not available by homology transfer. Today, arguably the core of the most powerful prediction methods for predicting RSA and other structural features of proteins is some form of Deep Learning, and all the state-of-the-art protein structure prediction tools rely on some Machine Learning algorithm. In this article we present a deep neural network architecture composed of stacks of Bidirectional Recurrent Neural Networks and Convolutional layers which is capable of mining information from long-range interactions within a protein sequence, and apply it to the prediction of protein RSA using a novel encoding method that we shall call "clipped". The final system we present, PaleAle 5.0, which is available as a public server, predicts RSA into two, three and four classes at an accuracy exceeding 80% in two classes, surpassing the performances of all the other predictors we have benchmarked.

Keywords Protein Structure Prediction, Solvent Accessibility, Evolutionary Information · Deep Learning · More

M. Kaleel, M. Torrisi, C.Mooney and G. Pollastri
UCD Institute for Discovery
School of Computer Science
University College Dublin
Belfield
IRL - Dublin 4
D04 V1W8
Tel.: +353 1 716 2620
E-mail: gianluca.pollastri@ucd.ie

1 Introduction

The relative degree of exposure to solvent molecules of amino acids in a protein (RSA) is an important one-dimensional (1D) property of proteins. Predicting 1D properties of proteins such as secondary structure (SS) [24], [14], [13] RSA and torsional angles [37], [36], [7] is a significant step towards solving the long standing biological problem of predicting a protein's 3D atomic co-ordinates [9] that can be used in protein function prediction, protein hydration position prediction [11], protein analysis, etc. Even though the Protein Data Bank (PDB) [8] regularly releases new protein structures determined by crystallography [32] or other experiments, the gap between the number of known and unknown structures is large and growing as experimentally solving protein structures is still expensive and time consuming. However, we have known for over half a century that the native conformation of proteins can be predicted solely based on their amino acid sequence [43] [6], and countless *in silico* methods have been developed over the following decades to produce inexpensive predictions of various structural aspects of proteins. In the specific case of RSA, linear regression models have been used [41], alongside neural network based regression [1], support vector machines (SVM) [27] [17], nearest neighbor algorithms [15], Bayesian probabilistic methods [39], random forests [31], information theory [26] and various other algorithms [21], in all cases relying on information gleaned from the increasing number of proteins whose structure has been determined by experimental means. As of today, almost all state-of-the-art RSA predictors use some form of Machine Learning, and increasingly Deep Learning [19]. RSA is typically predicted into two [22], three [42] or four classes [24], but also as the actual solvent accessible surface area [13]. In nearly all cases the absolute values of solvent accessibility, often obtained from the DSSP program [16], are normalized (hence the "Relative") into percentages of maximum exposed area calculated by different methods [38] [10] [34] [23] [40]. Current predictors can also be categorized as template based or *ab initio*, depending on whether they directly use information from homologues of known structure in the prediction process or rely exclusively on the sequence of the protein and, potentially, of evolutionary information in the form of sequences of other homologous protein of unknown structure. Even though template-based predictors are usually more accurate [28], they require one or more templates, making them not applicable in many cases. Selecting the right template can also be a significant challenge even when templates are found [25].

In this article we describe a new *ab initio* RSA predictor, PaleAle 5.0, that is a considerable improvement to our previous system PaleAle 4.0 [24] after a number of changes to the algorithms. We also describe tests we ran with different Machine Learning architectures, which led us to choose the final system composed of stacks of Bidirectional Recurrent Neural Networks (BRNN) [3] and Convolutional Neural Network layers (CNN) [20]. We performed all the training and testing in five-fold cross-validation on a very large, state-of-the-art redundancy reduced set containing over 15,000 experimentally resolved proteins. The final system is tested against a validation data set containing 1,601 proteins that are independent from the initial training/testing sets. RSA is predicted into 4-classes using solvent exposure thresholds of 4%, 25% and 50%, and re-cast into 2 classes with 25% exposure threshold or into 3 classes with 10% and 40% thresholds for comparison with other predictors. We compare PaleAle 5.0 with the recent predictors ACCpro

[22], RaptorX-Property [42], Spider3 [13] and PaleAle 4.0 [24] on our independent validation set.

We show that PaleAle 5.0 compares favourably with all *ab initio* competitors on all three formulations of the problem, roughly matching the template-based performances of ACCpro [22] in our tests. PaleAle 5.0 is freely available for academic users as a public web server and as a standalone program at `http://distilldeep.ucd.ie/paleale/`

Methods

Datasets

We extracted experimentally resolved publicly available data from the PDB [8] of December 2014 to form training and test sets for our models. The dataset is redundancy-reduced at a 25% identity threshold for sequence identity to avoid biases within sets and remove homologue pairs across training and testing sets. The solvent accessibility values are assigned by the DSSP [16] program from experimentally resolved 3D structures from the PDB. The target value (RSA) for each amino acid i in the final dataset RSA_i is calculated by the formula $RSA_i = SA_i/MAX_i \times 100\%$, where SA_i is the solvent accessibility of the i^{th} amino acid residue, in $\mathring{A}^2$, calculated by the DSSP program and MAX_i is the highest solvent accessibility of amino acid type i in $\mathring{A}^2$ [34]. The final dataset consists of 15,753 proteins containing 3,797,425 residues. These 15,753 protein sequences are then split into five subsets to perform five-fold cross-validation with more than 12,000 sequences in each training set and more than 3,000 sequences in each test set. Supplementary Figure 1 shows the percentage of amino acid residues below a given RSA threshold in our final dataset. Almost half of the residues in our final dataset are less than 20% exposed and some residues are more than 100% exposed. This issue arises from the fact that normalizing values commonly used (e.g. as calculated by Rose *et al* [34]) are the highest exposed area for an amino acid type based on available structures at the time of calculation, and that typically terminal amino acids are excluded from the calculations. While later research revealed that there are amino acids with higher exposure area for each amino acid type, older normalising values are retained for consistency and fair comparison with older predictors. The amino acids in our dataset are labelled into four classes using the following RSA ranges: $[0\% - 3\%], [4\% - 24\%], [25\% - 49\%]$ and $[50\% - \infty\%]$. The four classes were chosen to be roughly equally balanced as balanced classes are most informative. For comparison with other predictors, RSA is also re-cast into three and two classes. Ranges used for three class prediction are $[0\% - 10\%]$, $[11\% - 40\%]$ and $[41\% - \infty\%]$, and $[0\% - 24\%]$ and $[25\% - \infty\%]$ for two classes. Supplementary Table 4 shows the number of residues in each class for each classification problem in our sets.

We also constructed a validation set from the June 2017 PDB, completely independent of the datasets used to train and test our models, to estimate the final, unbiased performance of the resulting predictor. This validation set is redundancy reduced at 25% identity threshold for sequence identity within the set and against the training and test sets. This process resulted in 1,601 proteins and 352,864 amino acids. The validation set is labelled and enriched with alignments using the

same procedures and datasets that are employed to generate the training and test sets.

Input encoding

Evolutionary information

Encoding evolutionary information, usually in the form of frequency profiles is proven to increase the accuracy of models for predicting protein 1D properties [14] [35] [4]. Most modern *ab initio* RSA predictors use evolutionary information as an input. Sequence profiles for all sequences in our sets are generated from a combination of PSI-BLAST [2] and HHblits [33]. PSI-BLAST is run against the Jun 2016 version of UniRef90 [5] for three iterations with the e-value 0.001. HHblits is run against the February 2016 UniProt20 for three iterations with the e-value 0.001. These PSI-BLAST and HHblits profiles are combined to generate the overall alignments for each protein, and from these to generate frequency profiles used as the actual input to the models. We encoded these proteins into a single profile by calculating the frequencies of each amino acid type in each column of the resulting alignment. Each amino acid is encoded using 22 numbers consisting of the frequencies of the 20 common amino acids in the first 20 positions, the frequency of uncommon or unknown (B, J, O, U, X, Z) amino acids as the 21^{st} number and the frequency of gaps as the last number. If there are k occurrences of amino acid acid type t in a column i of an alignment, the frequency of amino acid type t in that position F_i^t is calculated simply as:

$$F_i^t = k/n$$

where n is the total number of amino acids occurring within column i.

The frequency of gaps for position i is calculated as:

$$F_i^{gap} = g/N$$

where g is the total number of gaps in the column and N is the total number of sequences, rather than amino acids, present in the alignment.
The same formulas are used to construct HHblits profiles using UniProt20 of February 2016. These plain frequencies are calculated with the assumption that every homologous sequence and its residues are equally important. However, some studies have shown that calculating the plain frequency is not the optimal solution [18].

Maximum Entropy profiles

Further to plain profiles, we also encoded profiles generated from PSI-BLAST and HHblits weighing aligned sequences by their entropy [18].

First F_i^t is calculated the same way as for plain frequency profiles. Then the weight of each homologous sequence W_s is calculated as the sum the entropies of its amino acids in each column of the alignment.

$$W_s = -\sum_{i=1}^{L} F_i^{s(i)} \log F_i^{s(i)} / L$$

where, L is the length of the sequence, and $s(i)$ is the type of amino acid occurring in sequence s in column i of the alignment.

The final 22 numbers encoding each position i are calculated by weighing each sequence s using the weight W_s. The weighed frequency of amino acid type t at position i is:

$$E_i^t = \sum_{s=1}^{S} W_s \times O_i^{s(t)} / \sum_{s=1}^{S} W_s \times O_i$$

where S is the total number of sequences in the alignment, $O_i^{s(t)}$ is 1 if amino acid t occurs in sequence s in column i of the alignment and 0 otherwise, and O_i is 1 if any valid amino acid (including unusual or unknown) occurs in sequence s in column i of the alignment and 0 if the sequence has a gap in that position.

The encoded frequency of gaps in this case is calculated as:

$$E_i^{gap} = \sum_{s=1}^{S} W_s \times O_i^{gap} / \sum_{s=1}^{S} W_s$$

where, again, S is the total number of sequences in the alignment, and O_i^{gap} is 1 if there is a gap in sequence s in column i of the alignment and 0 otherwise. That is, even in this weighed encoding amino acid frequencies are computed disregarding gaps when normalising, while gap frequencies are computed over the total number of sequences in the alignment.

It should be noted that this encoding scheme emphasises diversity, in that an unusual sequence (as measured by how infrequent its amino acids are with respect to the plain profile) will be weighed more than one mainly composed of amino acids commonly occurring in the profile.

Clipped encoding

As in previous research [35], [42], [30] the results from our tests show that encoding evolutionary information from alignments increases the accuracy of the prediction. However, classic encoding of an alignment (as a Position Specific Scoring Matrix, or as a frequency profile) conceals the identity of the amino acids present in the sequence S (the main sequence) from which the alignment is compiled. That is, while there is abundant evidence that evolutionary information is useful, its use typically also leads to some loss of potentially important information. To overcome this problem, we have tested a third encoding scheme, which we will call "clipped" encoding. In this scheme we first compute a frequency profile, plain or weighed, e.g. as described in the previous sections. Then, for each position in the alignment, we substitute the frequency of the amino acid that appears in that position in sequence S with the value 1, leaving the rest of the profile unchanged. It should be noted that no information from the initial (unclipped) profile is lost after this modification, as the frequency which we substitute with 1 could still be reconstructed as 1 minus the sum of the other frequencies. On the other hand, the identity of sequence S is

now present in the modified profile: its amino acids are those whose profile entry is 1.

Encoding the length

We also tested explicitly adding the sequence length to the input. This is obtained by adding a 23^{rd} number to the input. Specifically, the length is normalised by a fixed value (1000). It should be noted that in this case the 23^{rd} value of the input is identical for all positions of a sequence.

Clipped combined profile weighed by entropy

Sequence alignment tools find similar sequences based on the set of hypotheses underpinning the sequence alignment algorithm used. To mediate over these hypotheses, we tested an encoding to our system obtained by combining profiles generated using PSI-BLAST [2] and HHblits [33].

Predictive architecture

Stack of Bidirectional Recurrent Neural Networks and Convolutional Neural Networks

In order to intercept long-range signal in the sequence, we implemented a model made of a stack of Bidirectional Recurrent Neural Networks (BRNN) and Convolutional Neural Networks (CNN). In particular, the model has a first BRNN-CNN stage mapping the sequence into its RSA, and a second BRNN-CNN stage that filters the predictions of the fist stage.

In the BRNN we use [3], information about a whole sequence at a given position i is embedded in a forward memory F^i and a backward memory B^i, representing, respectively, the context to the left and to the right of position i.

The forward memory F^i at position i is generated via a forward transfer function as:

$$F^i = \phi(F^{i-1}, F^{i-2}, ..., F^{i-c}, I^i) \tag{1}$$

where I^i is the input in position i of the sequence, c is the length of the longest memory shortcut, and $\phi()$ is a non-linear function modelled by a 2-layer FNN.

Analogously, the backward memory B^i at position i is generated by a backward transfer function as:

$$B^i = \beta(B^{i+1}, B^{i+2}, ..., B^{i+c}, I^i) \tag{2}$$

Downstream of the BRNN, a CNN kernel takes as inputs a window of BRNN memories to map them into a local state O^i, i.e.:

$$O^i = \omega(F^{i-w}, ..., F^i, ..., F^{i+w}, B^{i-w}, ..., B^i, ..., B^{i+w} I_i) \tag{3}$$

Here $2w+1$ is the size of the kernel and $\omega()$ is a non-linear function implemented by a two-layered FNN.

The second, filtering BRNN-CNN stage is structurally identical to the first one, but its input is made of predictions by the first stage averaged over multiple contiguous windows, as in [29].

In particular, if O^i is the output of the first BRNN-CNN stage for position i in the sequence, the input to the second BRNN-CNN stage is the vector J^i:

$$J^i = (O^i, \hat{O}_w^{i-fc}, \ldots, \hat{O}_w^i \ \ldots, \hat{O}_w^{i+fc}) \tag{4}$$

where:

$$\hat{O}_w^v = \sum_{n=v-w}^{v+w} O^n/(2w+1) \tag{5}$$

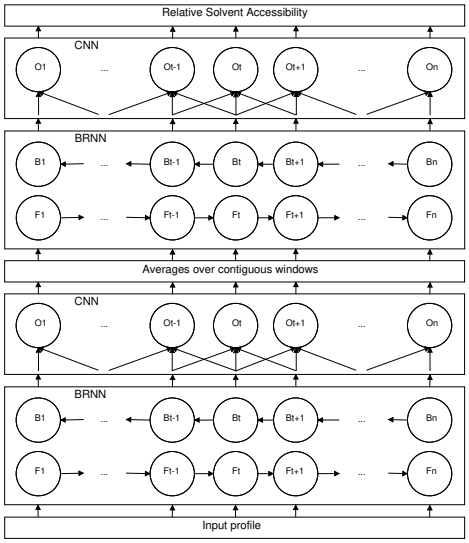

Fig. 1: Architecture of the double BRNN-CNN stack used in this work

We use $w = 7$, i.e. each value $\hat{O}_w^x$ is the average of first-stage predictions over a window of 15 amino acids centered at position x. We set $f = 2w + 1$, i.e. the windows on which the first-stage outputs are averaged are adjacent and non-overlapping. We fix $c = 10$, i.e. there are $2c+1 = 21$ such windows that concur to the input. That is, information extracted from up to $21 \times 15 = 315$ amino acids in total is presented at any position i to the second BRNN-CNN stage.

In summary, the sequence is processed by a first BRNN-CNN stage, and the outputs of the first stage are pooled into sets of averages which are processed by a second BRNN-CNN stage which is structurally identical to the first one apart from the different nature of the inputs. A diagram of the overall architecture is represented in Figure 1.

The first and second BRNN-CNN stages are supervised independently, but concurrently. That is, the first stage is supervised to predict RSA (rather than to produce a hidden representation) and so is the second stage, and the two stages are trained at the same time. In preliminary testing we have found this solution preferable to training the second stage using a fully trained first stage, possibly because the error surface for the second stage is not a static function of its internal weights and of the data, but morphs during the training process, which reduces the impact of basins of stagnating gradient.

We use $tanh()$ hidden units for all BRNN and CNN internal nodes, while we use softmax units for the overall outputs of both stages. It should be noted that the model we use is somewhat different from most off the shelf solutions in that, for instance, both BRNN transfer functions and CNN kernels are implemented as 2-layered FF neural networks rather as single layers. The BRNN transfer functions also contain shortcuts, i.e. model Markov chains with memory greater than 1, which create shorter paths for information propagation, similarly to Deep Residual Networks [12].

Training and Evaluation: We train eight different double BRNN-CNN stacks (details in Supplementary Table 1). The input to these stacks is encoded using clipped maximum entropy profiles with encoded length using alignments generated by PSI-BLAST, HHblits and a concatenation of the two. The models are ensembled and their performances measured on the unbiased validation set. The performances of individual models on the validation set and on all of the 5 cross-validation folds are reported in Supplementary Tables 2 and 3, while a comparison between first-stage and full stack results is reported in Supplementary Table 5.

To evaluate the performance of PaleAle 5.0 against other predictors, we use accuracy, macro averaged F1 score and individual class precision and recall. If true positives, false positives, true negatives and false negatives are, respectively tp, fp, tn and fn, then it is:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall}$$

Results and Discussion

Accuracy with different encodings

We tested our systems using frequency profiles extracted from alignments and without profiles. We generated different types of frequency profiles: plain, weighed based on sequence entropy, and using the "clipping" technique. We tested profiles generated using PSI-BLAST, HHblits and PSI-BLAST and HHblits concatenated. The performances of the system with different encoding techniques are listed in Table 1. The encoding schemes are tested using double BRNN-CNN stacks. Clipped profiles weighed by sequence entropy with sequence length encoded lead to the best performances, and so do profiles generated by both PSI-BLAST and HHblits combined (Table 1).

Performance compared to other systems

The comparisons with other systems are listed in the Table 2 and 3.

We compare our system with ACCpro [22], RaptorX-Property [42], PaleAle 4.0 [24] and Spider3 [13] using the same validation set of 1,601 proteins for a fair comparison. ACCpro has an *ab initio* and a template based version both predicting two-class RSA with a 25% threshold and the same normalizing values used by our predictor. In our tests ACCpro *ab initio* version has an accuracy of 76.70%

	BRNN-CNN	
Dataset	No-EL	EL
no profiles	46.65	46.76
plain	55.15	55.23
MaxEnt	55.13	55.17
Clip	56.10	56.20
Clip+MaxEnt	56.27	56.24
Clip+MaxEnt comb	56.81	56.93

Table 1: BRNN-CNN stack accuracy for four class RSA prediction on test set for different encodings. Plain are plain profiles, MaxEnt are profiles in which sequences are weighed based on their entropy, Clip are plain profiles with "clipping", Clip+MaxEnt comb are combined PSI-BLAST + HHblits profiles with clipping and entropy weighing. No-EL vs. EL are datasets without or with sequence length encoding added, respectively.

	2-Class		3-Class		4-Class	
System	ACC	F1	ACC	F1	ACC	F1
Consensus of all	79.9	0.80				
Consensus of top 3	80.3	0.80				
Accpro[a]	76.7	0.77	—-	—-	—-	—-
Accpro[t]	80.5	0.80	—-	—-	—-	—-
Spider3	77.9	0.78	61.2	0.62	49.0	49.0
RaptorX-Property[a]	—-	—-	55.5	0.54	—-	—-
RaptorX-Property[p]	—-	—-	63.3	0.63	—-	—-
PaleAle 4.0	78.2	0.78	—-	—-	52.5	0.52
PaleAle 5.0	80.5	0.80	66.4	0.66	56.5	0.56

Table 2: Performance of different predictors expressed as percentages of accuracy (ACC) and F1-Score (F1) on validation set. Consensus prediction is implemented by majority vote.

and macro averaged F1 score of 0.77 while ACCpro template-based has an accuracy of 80.5% and macro averaged F1 score of 0.80. RaptorX-Property is an *ab initio* predictor. RaptorX-Property predicts amino acids into three-classes, buried, intermediate and exposed, with 10% and 40% thresholds. RaptorX-Property has a version that does not use sequence profiles, RaptorX-Property[a], which predicts amino acids in our validation set into three-classes with an overall accuracy of 55.45% and macro averaged F1 score of 0.54 while RaptorX-Property[p], which uses profiles, has an accuracy of 63.25% and macro averaged F1 score of 0.63. However, RaptorX-Property uses a different set of values [23] to ours [34] to normalize absolute surface area into RSA. Therefore, we re-normalize RSA values in our validation set to [23] when testing RaprotX-Property for a fairer comparison. Spider3 [13] is an *ab initio* predictor using sequence profiles. Spider3 predicts overall exposed area of each amino acid in Angstroms (Å). This method gives us the freedom to use any normalizing values to convert overall exposed area into RSA, and to divide classes as required. Spider3 predicts all amino acids in our validation set with an accuracy of 77.91% and macro averaged F1 score of 0.78 for the two-class RSA problem with a threshold of 25%. For the three-class problem with 10% and 40% thresholds, Spider3's accuracy is 61.9% and macro averaged F1 score is 0.62, and 49.01% accuracy and macro averaged F1 score of 0.49 for the four-class problem with 4%, 25% and 50% thresholds.

PaleAle 5.0 predicts 56.4% of all residues in the correct class (a 3.8% improvement over PaleAle 4.0) in its 4-class version, 66.4% for the three class problem and 80.5% for the two-class problem. Precision and Recall of PaleAle 5.0, in general, are more balanced compared to other predictors. The accuracy of PaleAle 5.0 is almost equal to Accpro^{t}, however, Accpro^{t} is a template based predictor while PaleAle 5.0 is an *ab inito* predictor. Table 2 also shows the performances of 2-class consensus predictors between the top 3 and all predictors we have tested. The consensus roughly matches the results of PaleAle 5.0 and Accpro^{t} but does not improve on them.

Predictor	Precision				Recall			
	cls 0	cls 1	cls 2	cls 3	cls 0	cls 1	cls 2	cls 3
	Two class prediction							
Accpro^{a}	0.78	0.76	–	–	0.79	0.74	–	–
Accpro^{t}	0.81	0.80	–	–	0.83	0.78	–	–
Spider3	0.86	0.71	–	–	0.70	0.88	–	–
PaleAle 4.0	0.79	0.77	–	–	0.81	0.75	–	–
PaleAle 5.0	0.81	0.80	–	–	0.83	0.78	–	–
	Three class prediction							
Spider3	0.84	0.46	0.64	–	0.53	0.58	0.75	–
RaptorX-P^{a}	0.64	0.44	0.54	–	0.66	0.32	0.68	–
RaptorX-P^{p}	0.78	0.51	0.61	–	0.67	0.48	0.75	–
PaleAle 5.0	0.76	0.54	0.67	–	0.76	0.51	0.72	–
	Four class prediction							
Spider3	0.82	0.42	0.39	0.60	0.33	0.49	0.56	0.60
PaleAle 4.0	0.62	0.46	0.44	0.55	0.73	0.40	0.36	0.62
PaleAle 5.0	0.70	0.49	0.48	0.58	0.69	0.51	0.38	0.68

Table 3: Precision and recall of each class for two, three and four class predictions on validation set.

Discussion

The prediction of RSA alongside other 1D structural properties of proteins is often a fundamental step towards the prediction of protein structures. We have presented PaleAle 5.0 a predictor based on double BRNN-CNN stacks and a number of algorithmic improvements in the handling of evolutionary information in the form of frequency profiles. According to our tests the system performs at or above the state-of-the-art in the field, and is publicly available at `http://distilldeep.ucd.ie/paleale/`.

Acknowledgements The work of MK is supported by a grant from Irish Research Council [GOIPG/2014/603] and a UCD School of Computer Science Bursary. The work of MT is supported by a grant from the Irish Research Council [GOIPG/2015/3717].

The authors acknowledge the Research IT Service at University College Dublin for providing HPC resources that have contributed to the research results reported within this paper. http://www.ucd.ie/itservices/ourservices/researchit/

The authors declare that they have no conflict of interest. This article does not contain any studies with human participants or animals performed by any of the authors. No individual participant was included in this study, therefore no informed consent was necessary.

References

1. Adamczak, R., et al.: Accurate prediction of solvent accessibility using neural networks-based regression. Proteins: Structure, Function, and Bioinformatics **56**(4), 753–767 (2004). 10.1002/prot.20176
2. Altschul, S.F., et al.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research **25**(17), 3389–3402 (1997). 10.1093/nar/25.17.3389
3. Baldi, P., et al.: Exploiting the past and the future in protein secondary structure prediction. Bioinformatics **15**(11), 937–946 (1999). 10.1093/bioinformatics/15.11.937
4. Barton, G.J.: Protein secondary structure prediction. Current Opinion in Structural Biology **5**(3), 372–376 (1995). 10.1016/0959-440X(95)80099-9
5. Bateman, A., et al.: UniProt: the universal protein knowledgebase. Nucleic Acids Research **45**(D1), D158–D169 (2017). 10.1093/nar/gkw1099
6. Bering, C.L.: Biochemistry, Second Edition (Zubay, Geoffrey). Journal of Chemical Education **66**(3), A102 (1989). 10.1021/ed066pA102.2
7. Berjanskii, M.V., et al.: PREDITOR: a web server for predicting protein torsion angle restraints. Nucleic Acids Research **34**(suppl_2), W63–W69 (2006). 10.1093/nar/gkl341
8. Berman, H.M., et al.: The Protein Data Bank. Nucleic Acids Research **28**(1), 235–242 (2000)
9. Chan, H.S., et al.: Origins of structure in globular proteins. Proceedings of the National Academy of Sciences **87**(16), 6388–6392 (1990). 10.1073/pnas.87.16.6388
10. Chothia, C.: The nature of the accessible and buried surfaces in proteins. Journal of Molecular Biology **105**(1), 1–12 (1976). 10.1016/0022-2836(76)90191-1
11. Ehrlich, L., et al.: Prediction of protein hydration sites from sequence by modular neural networks. Protein Engineering, Design and Selection **11**(1), 11–19 (1998). 10.1093/protein/11.1.11
12. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
13. Heffernan, R., et al.: Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. Bioinformatics (Oxford, England) **33**(18), 2842–2849 (2017). 10.1093/bioinformatics/btx218
14. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices11edited by G. Von Heijne. Journal of Molecular Biology **292**(2), 195–202 (1999). 10.1006/jmbi.1999.3091
15. Joo, K., et al.: Sann: Solvent accessibility prediction of proteins by nearest neighbor method. Proteins: Structure, Function, and Bioinformatics **80**(7), 1791–1797 (2012). 10.1002/prot.24074
16. Kabsch, W., et al.: Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers **22**(12), 2577–2637 (1983). 10.1002/bip.360221211
17. Kim, H., et al.: Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3d local descriptor. Proteins: Structure, Function, and Bioinformatics **54**(3), 557–562 (2004). 10.1002/prot.10602
18. Krogh, A., et al.: Maximum entropy weighting of aligned sequences of proteins or DNA. Proceedings. International Conference on Intelligent Systems for Molecular Biology **3**, 215–221 (1995)
19. LeCun, Y., et al.: Deep learning. Nature **521**(7553), 436–444 (2015). 10.1038/nature14539
20. LeCun, Y., et al.: Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation **1**(4), 541–551 (1989). 10.1162/neco.1989.1.4.541

21. Li, X., et al.: New method for accurate prediction of solvent accessibility from protein sequence. Proteins: Structure, Function, and Bioinformatics **42**(1), 1–5 (2001). 10.1002/1097-0134(20010101)42:1¡1::AID-PROT10¿3.0.CO;2-N
22. Magnan, C.N., et al.: 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 (2014). 10.1093/bioinformatics/btu352
23. Miller, S., et al.: Interior and surface of monomeric proteins. Journal of Molecular Biology **196**(3), 641–656 (1987). 10.1016/0022-2836(87)90038-6
24. Mirabello, C., et al.: Porter, PaleAle 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. Bioinformatics **29**(16), 2056–2058 (2013). 10.1093/bioinformatics/btt344
25. Mooney, C., et al.: Beyond the Twilight Zone: Automated prediction of structural properties of proteins by recursive neural networks and remote homology information. Proteins: Structure, Function, and Bioinformatics **77**(1), 181–190 (2009). 10.1002/prot.22429
26. Naderi-Manesh, H., et al.: Prediction of protein surface accessibility with information theory. Proteins: Structure, Function, and Bioinformatics **42**(4), 452–459 (2001). 10.1002/1097-0134(20010301)42:4¡452::AID-PROT40¿3.0.CO;2-Q
27. Nguyen, M.N., et al.: Prediction of protein relative solvent accessibility with a two-stage SVM approach. Proteins: Structure, Function, and Bioinformatics **59**(1), 30–37 (2005). 10.1002/prot.20404
28. Pollastri, G., et al.: Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. BMC Bioinformatics **8**(1), 201 (2007). 10.1186/1471-2105-8-201
29. Pollastri, G., et al.: Porter: a new, accurate server for protein secondary structure prediction. Bioinformatics **21**(8), 1719–1720 (2005). 10.1093/bioinformatics/bti203
30. Pollastri, G., et al.: Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins: Structure, Function, and Bioinformatics **47**(2), 228–235 (2002). 10.1002/prot.10082
31. Pugalenthi, G., et al.: RSARF: Prediction of Residue Solvent Accessibility from Protein Sequence Using Random Forest Method. Protein and Peptide Letters **19**(1), 50–56 (2012). 10.2174/092986612798472875
32. Ramachandran, G.N.: Protein Structure and Crystallography. Science **141**(3577), 288–291 (1963)
33. Remmert, M., et al.: HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nature Methods **9**(2), 173–175 (2012). 10.1038/nmeth.1818
34. Rose, G.D., et al.: Hydrophobicity of amino acid residues in globular proteins. Science **229**(4716), 834–838 (1985). 10.1126/science.4023714
35. Rost Burkhard, et al.: Combining evolutionary information and neural networks to predict protein secondary structure. Proteins: Structure, Function, and Bioinformatics **19**(1), 55–72 (2004). 10.1002/prot.340190108
36. Shen, Y., et al.: Protein backbone and sidechain torsion angles predicted from NMR chemical shifts using artificial neural networks. Journal of Biomolecular NMR **56**(3), 227–241 (2013). 10.1007/s10858-013-9741-y
37. Shen, Y., et al.: TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts. Journal of Biomolecular NMR **44**(4), 213–223 (2009). 10.1007/s10858-009-9333-z
38. Shrake, A., et al.: Environment and exposure to solvent of protein atoms. Lysozyme and insulin. Journal of Molecular Biology **79**(2), 351–371 (1973). 10.1016/0022-2836(73)90011-9
39. Thompson, M.J., et al.: Predicting solvent accessibility: Higher accuracy using Bayesian statistics and optimized residue substitution classes. Proteins: Structure, Function, and Genetics **25**(1), 38–47 (1996). 10.1002/(SICI)1097-0134(199605)25:1¡38::AID-PROT4¿3.3.CO;2-H
40. Tien, M.Z., et al.: Maximum Allowed Solvent Accessibilites of Residues in Proteins. PLoS ONE **8**(11) (2013). 10.1371/journal.pone.0080635
41. Wagner, M., et al.: Linear Regression Models for Solvent Accessibility Prediction in Proteins. Journal of Computational Biology **12**(3), 355–369 (2005). 10.1089/cmb.2005.12.355
42. Wang, S., et al.: RaptorX-Property: a web server for protein structure property prediction. Nucleic Acids Research **44**(Web Server issue), W430–W435 (2016). 10.1093/nar/gkw306
43. White, F.H.: Regeneration of Native Secondary and Tertiary Structures by Air Oxidation of Reduced Ribonuclease. Journal of Biological Chemistry **236**(5), 1353–1360 (1961)