BIOINFORMATICS

Vol. 18 no. 6 2002
Pages 819–824

NETASA: neural network based prediction of solvent accessibility

Shandar Ahmad[1, 2, 3,*] *and M. Michael Gromiha*[2,†]

[1]*Institute of Multimedia and Software, Universiti Putra Malaysia, Serdang, 43400, Selangor, Malaysia,* [2]*RIKEN Tsukuba Institute, 3-1-1 Koyadai, Tsukuba, Ibaraki 305-0074, Japan and* [3]*Department of Biosciences, Jamia Millia Islamia, New Delhi, India*

Received on August 24, 2001; revised on November 2, 2001; December 22, 2001; accepted on January 7, 2002

ABSTRACT

Motivation: Prediction of the tertiary structure of a protein from its amino acid sequence is one of the most important problems in molecular biology. The successful prediction of solvent accessibility will be very helpful to achieve this goal. In the present work, we have implemented a server, NETASA for predicting solvent accessibility of amino acids using our newly optimized neural network algorithm. Several new features in the neural network architecture and training method have been introduced, and the network learns faster to provide accuracy values, which are comparable or better than other methods of ASA prediction.

Results: Prediction in two and three state classification systems with several thresholds are provided. Our prediction method achieved the accuracy level upto 90% for training and 88% for test data sets. Three state prediction results provide a maximum 65% accuracy for training and 63% for the test data. Applicability of neural networks for ASA prediction has been confirmed with a larger data set and wider range of state thresholds. Salient differences between a linear and exponential network for ASA prediction have been analysed.

Availability: Online predictions are freely available at: http://www.netasa.org. Linux ix86 binaries of the program written for this work may be obtained by email from the corresponding author.

Contact: shandar@jamia.net

INTRODUCTION

The accuracy of predicting the complete three dimensional structure of a protein is limited with the existing computational methods (Moult *et al.*, 1999). Secondary structure and solvent accessibility are, therefore subjects of interest in the field of structure prediction (Chandonia and Karplus, 1999). Neural networks have emerged to be a method of distinct choice for accurate prediction of these one dimensional properties of proteins (Qian and Senjowsky, 1988; Holbrook *et al.*, 1990; Rost and Sander, 1994). In this work, we propose an improved neural network method to predict solvent accessibility or accessible surface area (ASA) of amino acid residues in proteins. The algorithm has been implemented for an online prediction, using the server NETASA (http://www.netasa.org). The prediction results are found to be better than other methods in the literature.

MATERIALS AND METHODS

Network simulation

A feed forward neural network, consisting of an input, an output and a hidden layer has been designed for ASA training. The input layer successively reads binary inputs from an amino acid sequence database, and consists of 17 units of 21 bit binary vectors, when the prediction is being made for the central (9th) residue. Each of these 21 bit vectors represents the amino acid at the location being encoded. While coding these amino acids, each of the 21 bits are set to zero except the one which is assigned to a given amino acid type (20 amino acids +1 space for vacant/unknown position). All 17 units represent 8 neighbours on either side and the residue for which prediction has to be made. Thus a set of 357 binary inputs is provided as a single input for each residue location, in which a maximum of 17 bits will be set to 1 at one time. Choice of eight neighbours was made as it is now a well accepted number of neighbouring residues affecting protein conformation at a residue site (Manesh *et al.*, 2001).

Hidden layer and the output layers consist of the same number of units each as the number of accessibility states, n, desired in the output (e.g. two bits each in case of two state classification). Solvent accessibility is encoded by an n bit binary vector in which, all the coding bits are zero

*To whom correspondence should be addressed at: RIKEN Tsukuba Institute, 3-1-1, Koyadai, Tsukuba 305 0074, Ibaraki, Japan.
† Present address: Computational Biology Research Center (CBRC), AIST 2-41-6 Aomi, Koto-ku, Tokyo 135-0064, Japan.

Table 1. PDB codes of proteins used for training the network

1aba	1abr	1bdo	1beo	1bib
1bmf	1bnc	1btm	1btn	1cem
1ceo	1cew	1cfy	1chd	1chk
1cyx	1dea	1del	1dkz	1dos
1fua	1gai	1gpl	1gsa	1gtm
1hav	2i1b	2sns	3grs	3mdd

except one which corresponds to one of the accessibility states.

Training procedure

A set of 215 proteins with less than 25% homology and high resolution structures were selected for the present work. This set of 215 proteins is the same as the one recently used by Manesh *et al.* (2001) for implementing information theory to ASA prediction. This data is then divided into a training and test data sets. Only 30 proteins (7545 residues) were randomly selected for the training (Table 1). The remaining 185 proteins (42 037 residues) were kept in the test data set.

A large number of weights and biases need to be trained for the best accuracy levels with these proteins. Initialization of weights and biases was carried out by assigning them random values.

Training of weights is carried out one at a time, i.e. training one weight until the change in accuracy is less than a cutoff. Then we move on to the next weights and train them one by one. Presenting weights in the order of their occurrence may lead to over-training of weights in one region of weight space. This problem of unequal training of weights can be overcome by picking up the weights for training randomly from the whole set of weights. This will ensure that all network weights are equally trained. Several training cycles are run on the network and this further reduces the probability of unequal training. Starting with an appropriate value of cutoff, and gradually converging to a final cutoff value, is also helpful in checking this problem. After a certain number of training cycles, the accuracy cutoff is divided by a factor, and through a number of cycles, the network settles into the desired error minima. Training, one weight at a time, picked randomly, allows us to stop training anywhere. We evaluate prediction accuracy of the training data for every change in a weight, update that weight if there is an improvement in accuracy, or retain the value of that weight if there is no improvement in prediction accuracy. After a certain number of training cycles, prediction accuracy for the test data is evaluated. If the prediction accuracy on the test data does not improve in a cycle, training is stopped. We have also used linear activation function for the neurons, instead of the widely used sigmoidal function. This allows us to increase the sensitivity of the network over small changes in the weights.

Computation of solvent accessibility and prediction accuracy

Solvent accessibility (%) is defined as the ratio between the solvent accessible surface area of a residue in three dimensional structure and that in an extended tripeptide (Ala-X-Ala) conformation. The solvent accessible surface areas of all atoms have been computed using the program ASC (Eisenhaber and Argos, 1993) with the van der Waals radii of the atoms given by Ooi *et al.* (1987). The extended state coordinates have been computed using the ECEPP/2 algorithm (Momany *et al.*, 1975) with the dihedral angles of Oobatake and Ooi (1993).

Prediction accuracy for the training and test data sets is defined as the percentage of correctly predicted residues in the corresponding set of proteins. It assigns a negative score for both under-prediction and over-prediction.

RESULTS AND DISCUSSION

Prediction of ASA

The main NETASA server (http://www.netasa.org) for predicting solvent accessibility of each amino acid residue in a protein has been shown in Figure 1.

We have provided several thresholds (both for two and three states) for classifying residues as buried and exposed. As an example, we have selected the protein Human Thioredoxin (PDB code 1erv) for prediction, which was not included in the training data set. Two state prediction results for this protein for 25% threshold are provided in Table 2. Our method correctly predicts 89 out of 105 residues in buried or exposed category and the prediction accuracy is 83%, which is an excellent agreement between predicted and experimental ASA states obtained from its three dimensional structure (Vijaykumar *et al.*, 1987; Gromiha *et al.*, 1999).

Summary of all other prediction accuracy results for training and test sets of 30 and 185 proteins respectively, are presented in Table 3. A complete list of accuracy values for all training and test proteins can be seen through a link at http://www.netasa.org.

We found that NETASA could predict the solvent accessibility up to an accuracy of 90% for a 0% threshold of two state predictions. Further, the average accuracy lies between 70 and 90% for the two state predictions and 55 and 65% for three state predictions. These accuracy levels are superior to other methods available in the literature (see below).

There are several indices to measure the quality of ASA prediction, including single residue accuracy, Pearson's correlation coefficient and Matthews correlation coefficient (Matthews, 1975). However, for a two state

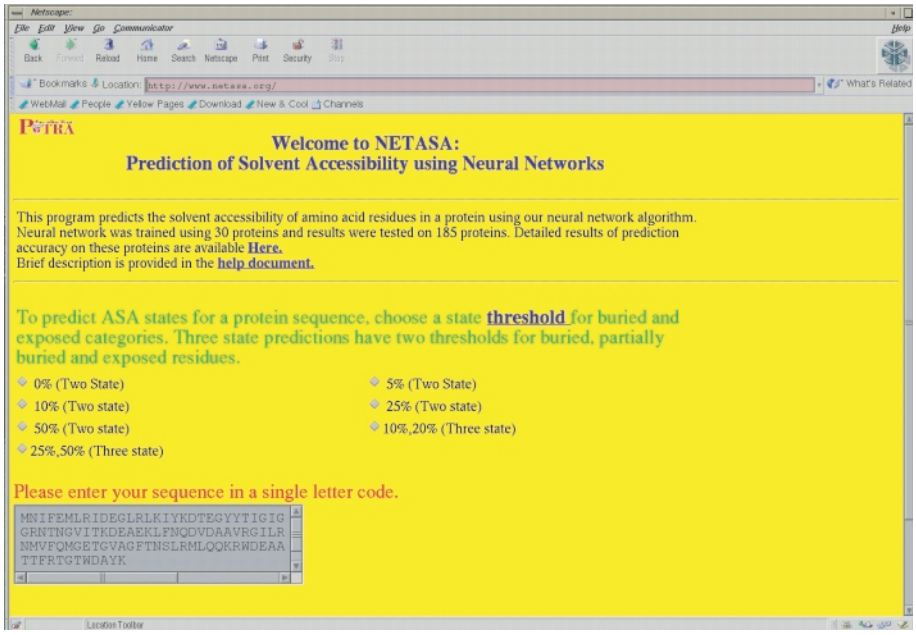

Fig. 1. NETASA web server for ASA prediction with an example to Human Thioredoxin (105 residues), PDB code 1ERV.

Table 2. Prediction results for Human Thioredoxin (PDB code 1ERV)

AA	M	V	K	Q	I	E	S	K	T	A	F	Q	E	A	L	D	A	A	G	D
Ex	e	b	e	e	b	e	e	e	e	e	b	e	e	b	b	e	e	b	e	e
Pr	b	b	e	e	b	e	e	e	e	b	b	e	e	b	b	e	e	e	e	e
AA	K	L	V	V	V	D	F	S	A	T	W	C	G	P	C	K	M	I	K	P
Ex	e	b	b	b	b	b	b	b	b	e	e	b	e	e	b	e	e	b	e	e
Pr	e	b	b	b	b	e	b	b	b	e	b	b	e	e	b	e	b	b	e	e
AA	F	F	H	S	L	S	E	K	Y	S	N	V	I	F	L	E	V	D	V	D
Ex	b	b	e	e	b	b	e	e	b	e	e	b	b	b	b	b	b	b	b	e
Pr	b	b	e	b	b	e	e	e	b	e	e	b	b	b	b	e	b	e	b	e
AA	D	C	Q	D	V	A	S	E	C	E	V	K	S	M	P	T	F	Q	F	F
Ex	e	b	e	e	b	b	e	e	b	e	b	e	e	b	b	b	b	b	b	b
Pr	e	b	e	e	b	b	e	e	b	e	b	e	e	b	e	b	b	e	b	b
AA	K	K	G	Q	K	V	G	E	F	S	G	A	N	K	E	K	L	E	A	T
Ex	e	e	e	e	e	e	e	e	b	e	e	e	e	e	e	e	b	b	e	b
Pr	e	e	e	e	e	b	e	b	b	e	e	b	e	e	e	e	b	e	e	b
AA	I	N	E	L	V															
Ex	b	e	e	b	e															
Pr	b	e	e	b	b															

AA = Amino Acid Residue, Ex = Experimental ASA state, Pr = Predicted ASA state. b = buried, e= exposed.

prediction, the two correlation coefficients mentioned here are identical. We have noticed that improvement in one of these indices of prediction quality does not necessarily imply a similar improvement in others. Neural network is capable of learning to maximize any index of prediction quality i.e. accuracy or correlation. Values in Table 3 are obtained by optimizing single residue accuracy values basically to enable a comparison with other methods. A training to maximize any of the correlation coefficients can however be similarly achieved.

Validation of results with re-partitioning the data

To validate the results obtained from the network mentioned above, we repeated all our calculations, this time,

Table 3. Summary of prediction accuracy and correlation for training and test datasets with 30 training proteins

State threshold (%)	Accuracy (%) and correlation*	
	Training	Test
0	89.8 (0.320)	87.9 (0.023)
5	76.1 (0.373)	74.6 (0.322)
10	75.2 (0.459)	71.2 (0.365)
25	73.1 (0.460)	70.3 (0.404)
50	80.1 (0.327)	75.9 (0.146)
10%, 20%	65.1 (0.417)	63.0 (0.373)
25%, 50%	60.9 (0.348)	55.0 (0.229)

(*) Values in brackets represent correlation coefficients.

Table 4. Summary of prediction accuracy and correlation for training, test and validation datasets after re-partitioning the data

State threshold (%)	Accuracy (%) and correlation*		
	Training	Test	Validation
0	88.8 (0.320)	89.2 (0.071)	88.1 (0.097)
5	75.3 (0.356)	73.7 (0.251)	72.1 (0.240)
10	73.1 (0.413)	71.3 (0.352)	71.0 (0.325)
25	72.6 (0.455)	71.0 (0.414)	71.1 (0.414)
50	76.7 (0.134)	74.7 (0.117)	75.1 (0.113)

(*) Values in brackets represent correlation coefficients.

dividing the 215 proteins into three sets. One of the data sets was used for training, others to determine where the training be stopped (just the same way as described above) and the third dataset was used to validate the results after the training has been completed. This leaves 72, 72 and 71 proteins in each data set. Training, test and validation data were rotated for all possible six combinations, and the average accuracy for training, test and validation data were obtained. Using this scheme, we could reproduce all the accuracy values mentioned in Table 3, with the exception of 5% threshold where, the resultant accuracy was found to be 72.1%. Accuracy and correlation results of such predictions are summarized in Table 4. It is also observed that a higher accuracy in the extreme threshold states was accompanied by poorer correlation coefficients. This suggests that the network has a tendency to over-predict higher populated states. To examine this aspect, the 50% threshold predictions, which showed a 75% accuracy, were retrained for best correlations. Training, test and validation data accuracy with correlation optimizations, are found to be 74.0, 71.6 and 71.1% respectively. Correlation coefficients corresponding to these values are 0.332, 0.273 and 0.251 respectively. Similar training for other thresholds indicates that the prediction accuracy for the best correlations will vary in a narrow range of 71–72% for all state thresholds.

Comparison with other methods

There have been several attempts in the recent past to predict accessibility of amino acids from sequence, with an objective to reduce the gap between the number of known sequences and known three dimensional structures. A direct comparison of these methods is not possible owing to the reasons such as arbitrarily different choices of state thresholds for ASA classification and use of different methods to calculate ASA. However, a general comparison of the reported accuracy values was shown to have predictions close to 73% in the two state model and 58% in the three state model (Richardson and Barlow, 1999). Some more investigations have been reported after this review (Giorgi *et al.*, 1999; Carugo, 2000; Li and Pan, 2001; Manesh *et al.*, 2001). Giorgi *et al.* (1999) have used a knowledge based prediction model and reported accuracy values of 85.0, 77.0 and 70.7% for a two state model with 0, 9 and 25% state thresholds. However, their prediction results report several residues as 'unknown', and hence comparison of these figures with other methods, including the present work is not possible. Further, predictions in PredAcc (Giorgi *et al.*, 1999) provide a two state model only as compared to two and three state models in the present work. Carugo (2000) also developed an independent method for ASA prediction and could obtain an accuracy of 68.7% for a two state model. Li and Pan (2001), also used a somewhat similar method and could obtain a prediction accuracy of about 71.5%. Results for a three state prediction by their method were not reported. Cuff and Barton (2000) have reported a maximum 86.7% accuracy for a 0% threshold with the inclusion of sequence alignment information. A comprehensive study using information theory was reported by Manesh *et al.* (2001). In their work, they have applied information theory to several state models including two and three states. The most significant results relating to two state models produce an accuracy of about 70% and for a three state models the accuracy is 53 and 58% for two choices of thresholds. They have also used their own method to calculate ASA values from PDB data, using these figures of accuracy which seem to be higher. However, for this comparison, we take those values obtained using DSSP as that is a more widely accepted method of ASA calculation. In view of the accuracy figures mentioned above, the accuracy values in the present work are evidently better than other methods reported in the literature (Table 3).

A simpler choice of network design and training

Activation functions and network biases In a multilayer neural network, the neural signals are propagated using an activation function, which collects all the excitatory inputs from the previous layer, calculates a weighted sum, transforms it via a linear, sigmoidal or exponential

function and propagates the resulting signal to the next layer. In a two state prediction, this activation is calculated only for two nodes in the hidden layer. In a linear network the final state of the two output nodes are $w_1a + w_2b$ and $w_3a + w_4b$ respectively, where a and b are the activations received by the nodes in the hidden layer from the input layer and w's are the weights connecting hidden layers and the output layer units. In an exponential activation, the status of the output nodes will be altered to $w_1e^a + w_2e^b$ and $w_3e^a + w_4e^b$ respectively. The actual prediction is made by subtracting the status of one unit from the other. Positive and negative values of this difference (D) determine, if the residue is predicted as buried or exposed respectively. It is obvious that the value of D for a linear network (D_l) and for an exponential network (D_e), do not become negative or positive at the same time. This means $D_l > 0$ does not necessarily imply D_e will also be greater than zero. We examined this relationship in the predictions and found that the networks trained for linear activations, may give as much as 70% lower accuracy if the same weights were used for an exponential network. However, the net accuracy levels achieved in the two designs of the network were not found to differ significantly, and therefore, we conclude a simpler linear network can also be used to obtain similar prediction results as would be obtained by exponential or sigmoidal functions.

We also observe that training of network biases is a relatively redundant exercise in this problem. Since, neural network is an approximation method anyway and many simultaneous solutions exist for the kind of training, we need, we experimented by setting all network biases to zero and found that an exclusive training of weights, with no network biases, achieves identical accuracy levels as the ones with non-zero biases and training of biases and weights together. Thus, doing away with network biases is a simplification, which can be afforded in this problem.

Effect of window size and observation of over-training It was found that the window size of the input sequence data, does not affect the accuracy levels significantly starting from a window size of 3 and going upto 8. This confirms the results previously reported by Rost and Sander (1994). To include all possible sequence information, we preferred a window size of eight residues. However, for a training data with a relatively high state threshold, a window size as large as 8 creates a network too large for the training data available in each state. For example for an 85% threshold, there are just 1745 residues in the exposed state for whole data set. Dividing it into training and test data sets will leave approximately 600 residues only. A network with eight neighbours has 718 weights and therefore training such a large network for smaller data is not justified. We therefore reduce the network size to allow having a number of residues in each prediction state which is at least four times the number of weights. Using this as a thumb rule, we could train networks for higher thresholds. We find that the accuracy levels with the data partitioning into three sets, gives similar accuracy levels for higher thresholds also, although the window size could be as low as three.

Rost and Sander (1994) have reported that the prediction accuracy for their training and test data sets is the same (71.4%). Significantly, though it has been proved that a multi layer neural network can be trained to any degree of accuracy, and if the training is stopped, it should be done, when test data accuracy has started falling with an improvement in training data accuracy (Hornik *et al.*, 1989). In the present work, we do observe over-training. As an example, for the case of 5% threshold, we reported here 76.1% accuracy for training and 74.6% for test data sets. We actually tried to train this network further and could get a 77.9% accuracy for training data, but it reduced the test data accuracy to only 71.9%. So, in our network, we have clearly observed an over-training. One important result we observed is that the incidence of over-training becomes more likely for larger window sizes, as the maxima of test data accuracy is then observed at relatively higher values of training prediction accuracy.

Possible reasons for better accuracy In the present work, we find significantly better accuracy values specially in the extremely populated states. These high values of accuracy are accompanied by relatively low correlations. The following possible reasons for better accuracy values emerge.

The network has been trained to maximize accuracy scores, which means that the network may have over-predictions in higher populated states. In this situation correlation coefficients may be regarded as a better candidate of prediction quality. Previously reported networks do not provide information on correlations in all thresholds and therefore, we cannot conclusively say if a similar observation was made in these networks. However, at least two authors have reported observation of highest prediction accuracy in the extreme state calssification of 0% threshold (Cuff and Barton, 2000; Giorgi *et al.*, 1999). This seems quite likely that these high accuracy values were accompanied by low correlation coefficients.

The other possibility could have been that the relationship between the residues and their accessibility may be better represented by a linear network rather than a sigmoidal network. However, when we train the network for a sigmoidal activation and analyse this possibility, we do not find significant differences in the two ways of training except in the speed at which the two networks learn.

CONCLUSION

We propose a simplified neural network to predict the solvent accessibility of amino acid residues in a protein from its primary sequence. This procedure includes a linear activation function and some changes in the training procedure. Together, this provides accuracy levels equally good and even better than other existing methods. Hence, we provide an alternative approach to using neural networks for ASA prediction and also confirm the application of this method with a wider range of thresholds and larger data sets.

ACKNOWLEDGEMENTS

We would like to dedicate this work to S.A.'s mother who passed away during its progress.

REFERENCES

Carugo,O. (2000) Predicting residue solvent accessibility from protein sequence by considering the sequence environment. *Protein Eng.*, **13**, 607–609.

Cuff,J.A. and Barton,G.J. (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins*, **40**, 502–511.

Chandonia,J. and Karplus,M. (1999) New methods for accurate prediction of protein secondary structure. *Proteins*, **35**, 293–306.

Eisenhaber,F. and Argos,P. (1993) Improved strategy in analytical surface calculation for molecular systems—handling of singularities and computational efficiency. *J. Comp. Chem.*, **14**, 1272–1280.

Gromiha,M.M., Oobatake,M., Kono,H., Uedaira,H. and Sarai,A. (1999) Role of structural and sequence information for predicting protein stability changes: comparison between buried and partially buried mutations. *Protein Eng.*, **12**, 549–555.

Holbrook,S.R., Muskal,S. and Kim,S.H. (1990) Predicting surface exposure of amino acids from protein sequence. *Protein Eng.*, **3**, 659–665.

Hornik,K., Stinchocombe,M. and White,H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks*, **2**, 359–366.

Giorgi,M.H.M., Hazout,S. and Tuffery,P. (1999) PredAcc: prediction of solvent accessibility. *Bioinformatics*, **15**, 176–177.

Li,X. and Pan,X.M. (2001) New method for accurate prediction of solvent accessibility from protein sequence. *Proteins*, **42**, 1–5.

Manesh,H.N., Sadeghi,M., Arab,S. and Movahedi, (2001) Prediction of protein surface accessibility with information theory. *Proteins*, **42**, 452–459.

Matthews,B.W. (1975) Comparison of predicted and observed secondary structure of T_4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Moult,J., Hubbard,T., Fidelis,K. and Pedersen,J.T. (1999) Critical assessment of methods for protein structure prediction (CASP): round III. *Proteins*, **37**, 2–6.

Momany,F.A., McGuire,R.F., Burgess,A.W. and Scheraga,H.A. (1975) Eneregy parameters in polypeptides. 7 Geometric parameters, partial atmoic charges, non bonded interactions, hydrogen bond interactions and intrinsic torsional potentials for naturally occuring amino acids. *J. Phys. Chem.*, **79**, 2361–2381.

Ooi,T., Oobatake,M., Nemethy,G. and Scheraga,H.A. (1987) Accessible surface areas as a measure of the thermodynamic parameters of hydration of peptides. *Proc. Natl Acad. Sci. USA*, **84**, 3086–3090.

Oobatake,M. and Ooi,T. (1993) Hydration and heat stability effects on protein unfolding. *Prog. Biophys. Mol. Biol.*, **59**, 237–284.

Qian,N. and Senjowsky,T.J. (1988) Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.*, **202**, 865–884.

Richardson,C.J. and Barlow,D.J. (1999) The bottom-line for prediction of residue solvent accessibility. *Protein Eng.*, **12**, 1051–1054.

Rost,B. and Sander,C. (1994) Conservation and prediction of solvent accessibility in protein families. *Proteins*, **20**, 216–226.

Vijaykumar,S., Bugg,C.E. and Cook,W.J. (1987) Structure of Ubiquitin refined at 1.8 A resolution. *J. Mol. Biol.*, **194**, 531–544.