Predicting the structure of protein fragments.

Kristofer Hallen

31st January 2002

Abstract

This project tries different methods to predict structural similarity between protein fragments. The methods used were a simple dot product, an artificial neural network (ANN) and a support vector machine (SVM). Different types of input were used, amino acid sequences, PSI-BLAST profiles and predicted secondary structures. The performance of the methods were compared and the two learning methods, ANN and SVM performed better than the dot product method. Contrary to earlier studies type of input didn't affect the performance.

Contents

1 Introduction

This project concerned the three dimensional structure of proteins, an important part in finding the function of a protein. Experimental methods can be both expensive and time consuming and therefore methods are being developed to determine the structure with computers. Proteins are constructed by joining amino acids through peptide bonds into long chains. The chain is folded into a three dimensonal structure and this study has examined the possibility to predict the similarity between fragments based on the amino acid sequence.

Proteins have both secondary and tertiary structure where the tertiary structure is the full 3D structure. The secondary structure is a local structure with three main types, helix, sheet and coil. The methods for predicting the secondary structure, for example with neural networks, are performing well with a correctness of over 70%, this can give us information on the function of the protein and be a step towards finding the tertiary structure.

The idea of trying to predict the structure of small fragments has already been tried and the group of David Baker at the University of Washington has been able to put fragments with known structure together into a full size protein [1], 20% correct. The structure is determined by the sequence and although interactions between distant residues is important for the overall structure of the protein, there is also a relationship between local sequence and local structure [2].

The project was conducted at Stockholm Bioinformatics Center (SBC) under supervision of Arne Elofsson and was part of my year at the Stockholm Graduate School of Molecular Life Sciences.

2 Theory

The two main methods used were artificial neural networks and support vector machines. Both are established machine learning methods and a lot of implementations exist. Both methods also need to be trained (learning) before they can be used on unknown data.

2.1 Artificial neural networks

Learning methods should produce better results than the dot product. Artificial neural networks imitate the way biological neural networks work. It's a network of nodes, each node has three main features:

1. The synaptic weight. Each input signal is multiplied with a synaptic weight before it reaches the summing junction.
2. The summing junction adds all incoming signals.
3. An activation function that limits the output to some value.

An external bias is also added, it decreases or increases the net input to the activation function. The network used here is a multilayer feed forward network. The input data is given to a layer of nodes that process the data and their output is the input for the next layer of neurons, these are hidden neurons and you only see the output from the last layer. The way the network learns is by adjusting the synaptic weights. This is done using the back-propagation algorithm. First, the information is passed trough the net and output is produced. Then, in the backward pass, the synaptic weights are optimized so that the output of the network becomes closer to the known correct answer.

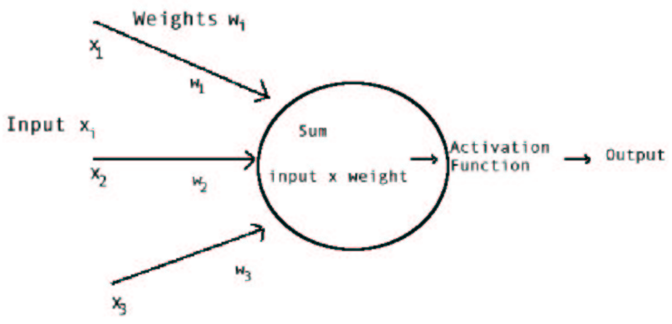

Figure 1: The neuron in the ANN. Inputs are multiplied with the weights and added up. The sum is sent to the activation function.

2.2 Support vector machines

SVM is another learning method and operates by finding a hypersurface in the space of possible inputs. This hypersurface will attempt to split the positive examples from the negative examples. Given l observations where each observation consists of a pair: a vector $x_i \in R^n$, i=1,...,l and a value y_i that tells if it is a true example or a false. Here the vector $\mathbf{x}$ is the vector with the encoded sequence (see 3.2) and the y is either 1 or 0 for similar or non similar. It is assumed that there exists some unknown probability distribution P(x,y) from with the data is drawn. The support vector machine is a machine that we want to learn the mapping $\mathbf{x} \rightarrow y$. The machine is defined by a set of mappings $\mathbf{x} \rightarrow f(\mathbf{x}, \alpha)$ and when we train the machine we adjust the variables α, trying to find the function that gives the smallest error (compare with changning the synaptic weights in the neural network).

3 Methods and data format

3.1 The fragments

The data used was amino acid sequencies for proteins. The sequences were fragmented, divided into smaller chains. Each fragment has a length, fragment size, l. But the amino acids on each side of the fragment could also contain information about similarity and therefore these can also be given to the programs. The number of adjacent residues together with the fragment is called the input window size.

3.2 Sequence data

To put the sequence data into the computer it needs to be coded. Several methods to do this exist but the one used here is sparse encoding scheme. This takes an amino acid sequence and each residue is encoded by a vector of 20 elements:

ALA: 10000000000000000000
CYS: 01000000000000000000
ASP: 00100000000000000000
.
.

More information can be used when encoding the sequence. Proteins with similar sequencies usually have similar structures so instead of just using a single amino acid in a given position we can look at how often different aminoacids appear in that position. A version of BLAST[4], Position-Specific-Iterated BLAST, PSI-BLAST[3] can give this information. PSI-BLAST searches a protein sequence database with a query sequence and constructs a multiple alignment and a profile with the frequencies of amino acids.

Example, a high value indicates that the amino acid is common at this position:

Pos	A	R	N	D	C	Q	E	G	H	I	L	K	M	F ...
1	-2	0	6	0	-3	0	0	0	0	-3	-3	0	-2	-3 ...

3.3 Including secondary structure predictions

The secondary structure predicition can also be used in the learning methods. Each amino acid can be either helix, encoded as the vector, sheet or coil. These were also encoded as vectors:

Helix	100
Sheet	010
Coil	001

The secondary structure prediction vector was inserted into the PSI-BLAST sequence after each amino acid.

3.4 Methods

Different methods have been used here to predict similarity between protein fragments, artificial neural networks (ANN), support vector machines (SVM) and a dot product. The ANN and SVM are learning methods. They are given a dataset for training, this contains positive and negative examples and the correct answer for each example. The methods need to be trained before they can be used on unknown data.

3.5 Dot Product

One of the simplest ways of comparing to compare two protein fragments is to make a dot product between the two vectors consisting of the combined amino acid vectors (see section 3.2) for each sequence. Similar sequencies should give a high and nonsimilar a low dot product. The pairs with a dot product above a certain cut off were selected as similar fragments. Baker [1] finds similar structures by doing at PSI-BLAST search and selecting proteins with a scoring function biased, but not constrained, to the secondary structure prediction.

3.6 Measuring the performance

3.6.1 MCC

The Mathews Correlations Coefficient (MCC) is a measure used to measure the performance of a predicting machine. It is better than percent of correctly classified examples when the datasets differ alot in number of examples.

$$MCC = \left(\frac{pn - ou}{\sqrt{(p-o)(p+u)(n+o)(n+u)}} \right)$$

where:

- p is the number of correctly classified similar fragments.
- n is the number of correctly classified non similar fragments.
- o is the number of non similar fragments incorrectly classified as similar fragments.
- u is the number of similar fragments incorrectly classified as non similar fragments.

MCC is a value between -1 and 1 where a 1 means a fully perfect prediction and -1 means a fully imperfect prediction. A MCC of 0 means that there is no correlation at all.

3.6.2 Sensitivity and Specificity

- Sensitivity is a value of how many of the similar fragments that the method finds.
  $Sensitivity = \frac{p}{p+u}$
- Specificity is a value of how many of the fragments, classified as similar, that are similar.
  $Specificity = \frac{p}{p+o}$

4 Implementation

The goal was to tell if two protein fragments are similar. This needs the input data needed to be formated, get a prediction and output data to be handled. Programs, in Perl, formated the input data and run the prediction programs. The perl programs were based on a program by Christian Ottosson [9]. The neural network was the Netlab [5][6] library for Matlab [10]. The support vector machine was SVM^{light}[7] a SVM implementation i C.

The datasets were randomly selected proteins from the SCOP [11] [12] classes 1 (all alpha), 2 (all beta), 3 (alpha and beta) and 4 (alpha and beta). The data was divided into training and validation sets. The proteins were fragmented and all fragments were used resulting in 10000 to 20000 fragments. Ten different validation set - training set pairs were used in the comparisons of the methods.

Twelve different validation set - training set pairs with were also constructed, six with proteins from only class 1 and six with proteins from class 2.

To train and validate the methods we need the similarity between two fragments, this was defined as the root mean square deviation (RMSD) between the

C_α coordinates for the two fragments. This was calculated by a RMSD program using the Protein Data Bank (PDB)[8] files for the protein and was turned into a discret value, 1 if the RMSD was below the cutoff (similar) and 0 if it was above the cut off (non similar). The cut off used was 1 Å.

To train the SVM and ANN effectively the training sets needed to be balanced, the balance is the number of negative examples divided with the number of positive examples. The balance used for the training sets was 3.

The balance in real data is much higher and since it turned out that the number of positive examples was even lower for proteins with a lot of beta sheets the validation sets were also balanced to give more positive examples. The validation set balance was 10.

The fragment size was 9 residues and the window sizes from 9 to 25 with steps of two were tried. A window size of nine means that no residues outside the fragment was used and 25 means that 8 residues on each side of the fragment was used. The prediction didn't improve with a bigger window size and the final comparisons of the methods was made with a window size of 9.

Some fragments were very similar and appeared a lot in the positive examples. To not give a bias for these in the training a constraint was introduced, a fragment couldn't appear in more than three positive examples.

Fragment size	9
Window size	9-25
Balance, training set	3
Balance, validation set	10
RMSD cut off	1 Å

Table 1: Implementation data.

5 Results

Method	data	MCC	SD	Sens	Spec
dotp	profile	0.09	0.02	0.16	0.30
ANN	sequence	0.23	0.08	0.45	0.24
ANN	profile	0.26	0.07	0.47	0.26
SVM	profile	0.24	0.09	0.19	0.42
ANN	profile + sec.struct.	0.28	0.07	0.55	0.26

Table 2: Mean values, MCC, standard deviaton, Sensitivity and Specificity for different methods and different input data. Sequence is amino acid sequence, profile is PSI-BLAST profile.

5.1 Dot Product

The dot product was simple and ran fast. Only the PSI-BLAST data input was used and different cut offs were tested and the best one was selected. The method didn't perform well, the mean MCC was 0.09 ± 0.02 and it didn't improve with a bigger window size, see Table 2 and figure 2

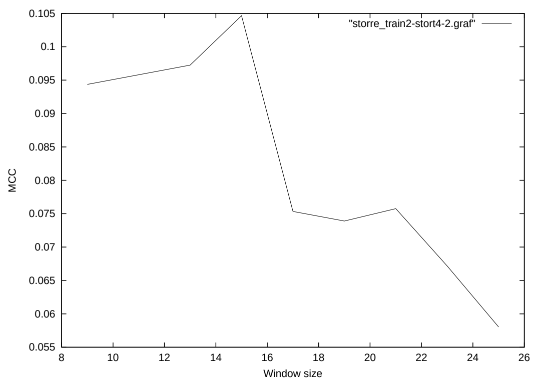

Figure 2: Dotproduct, MCC for different window sizes

5.2 Neural network

The neural network had a mean MCC that varied very little between the different inputs (Table 2.) Several different inputs were tested:

- Sequence, only the amino acid sequence. MCC 0.23±0.08
- Profile, the PSI-BLAST profile. MCC 0.26±0.07
- Profile and secondary structure prediction. MCC 0.28±0.07

The performance didn't increase with a bigger window (figure 3).

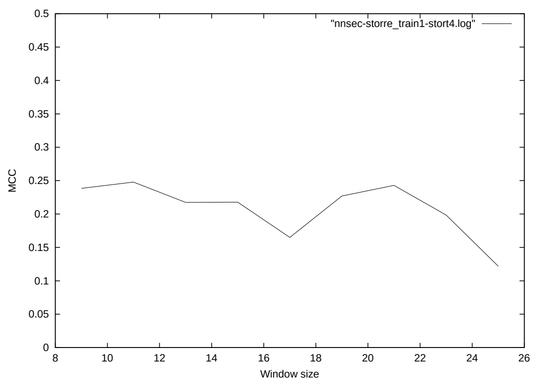

Figure 3: Neural network, MCC for different window sizes

5.3 Support vector machine

The input to the support vector machine was the PSI-BLAST profile as input and the MMC was 0.24±0.09. The performance didn't increase with window size here either.

5.4 Helixes and sheets

To see if there was a difference in how easy it was to find similar fragments for proteins with a lot of helixes and proteins with a lot of sheets two data sheets.

The data sets with only helixes and only sheets were tested on the ANN and the SVM. The performance of the two methods did not differ much and they both showed the same pattern, predicting helixes was better than for the mixed datasets with MMC of 0.44 (ANN) and 0.48 (SVM), table 3) The MCC for the sheets was zero.

Method	data	MCC	stdv	Sens	Spec
ANN	profile, helix	0.44	0.04	0.57	0.55
SVM	profile, helix	0.48	0.03	0.69	0.52
ANN	profile, sheet	0.02	0.03	0.18	0.10
SVM	profile, sheet	0			

Table 3: Proteins with only helixes and only sheets.

6 Conclusion

The learning methods are clearly better than the simple dotproduct method. The SVM and ANN performance is similar. The different inputs gave a slightly higher mean MCC when including the secondary structure prediction but the standard deviation is so large (0.07) that it can't be said to be better.

The problems seem to arise when trying to predict fragments from proteins containing a lot of sheets. The performance for helixes is almost twice that of the mixed data sets. Instead of comparing all the fragments it would probably be better to build a library with frequently occuring fragment structures. We can find similar fragments in proteins containing a lot of sheets but it's time consuming and they disappear in all the non similar fragments.

It is probably also easier to find structures for fragments that are only in a helix or a sheet, fragments on border lines can differ much more. Maybe the secondary structure prediction could come in handy there, you don't fragment the unknown protein randomly but in different secondary structures. Or you could try to fragment the protein several times (if you don't want do do all possible combinations) and see if the performance changes.

The neural networks and the SVM didn't differ much in time used for the computations but the SVM seemed to need more memory. This of course depends on the computers and the form of networks used but 50-70 proteins each for the training and validation seemed to be the limit.

References

[1] Simons KT, Bonneau R, Ruczinski I I, Baker D. Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins*, 1999;37(S3):171-176

[2] Bystroff C, Simons K, Han K, Baker D. Local sequence-structure correlations in proteins. *Current opinion in Biotechnology*, 7:417-421,1996.

[3] S.F Altschul, T.L Madden, A.A Schäffer, J.Zhang, Z.Zhang, W Miller, and D.J Lipman Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389-3402,1997.

[4] S.F Altschul, W.Gish, E.W. Myers and D.J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403-410, 1990.

[5] I.Nabney. Netlab software. http://www.ncrg.aston.ac.uk/netlab/

[6] I.Nabney. *Netlab. Algorithms for Patterns Recognition.* Advances in Pattern Recognition. Springer-Verlag, Heidelberg, 2001.

[7] T. Joachims. SVM^{light} software. http://svmlight.joachims.org/

[8] Research Collaboratory for Structural Bioinformatics. The RSCB Protein data Bank. http://www.rcsb.org/pdb/

[9] Christian Ottosson. Prediciting the structure of protein fragments. Masters's thesis. NADA, KTH. 2001.

[10] MathWorks. Matlab. http://www.mathworks.com/products/matlab/

[11] SCOP, structural classification of proteins. http://scop.mrc-lmb.cam.ac.uk/scop/

[12] Murzin A.G, Brenner S.E, Hubbart T. Chothia C. *J. Mol. biol* 247, 536-540. 1995.