

Prediction of Protein Domain-Types by Backpropagation

János Murvai¹, Csaba Szepesvári¹, Csanád Bachrati⁴ and Sándor Pongor^{2,3}

¹ MTA-JATE Research Group on Artificial Intelligence,
Szeged 6720, Aradi vrt. tere 1. Hungary

² ABC Institute for Biochemistry and Protein Research, 2100 Gödöllő, Hungary

³ International Centre for Genetic Engineering and Biotechnology,
Area Science Park, 34012 Trieste, Italy

⁴ Institute of Genetics, Biological Research Centre, Hungarian Academy of Sciences,
Szeged, POB. 521, H-6701, Hungary

Abstract. We propose a neural net solution for the recognition of the domain-types of proteins, which is a hard and important problem in biology. We have found that using a clever preprocessing technique relatively small neural networks perform surprisingly well. The performances of the neural nets were measured by cross-validation and Hoeffding's inequality was utilized for the estimation of a confidence interval of the estimates.

1 Introduction

Genome projects produce a huge amount of genetic sequence data. The ultimate goal is to assign a biological function to these data, the natural approach being to translate the data into protein sequences and assign the biological function of the protein to the data. However, when unclassified genetic data is translated proteins with unknown biological function may occur. Thus an important task is the recognition of the biological function of arbitrary protein sequences.

At present, the function assignment cannot be fully automated. The usual method works as follows: If two protein sequences share a common subsequence which is longer than 80% of the length of the sequences then the biological function of the two proteins is identified. Thus by organizing large amounts of proteins with known functions into databases, functions can be almost automatically assigned to proteins. This preliminary identification is carried out with standard database-search programs such as FASTA [17,12], or BLAST3 [13,1]. An estimated 25 percent of the new sequences have much less identity, so a function has to be assigned by human evaluation. The task is formidable since the number of data is far greater than that which can be processed by database annotators.

The hardness of the task arises from the biological nature of the problem. If two large proteins share only a few common segments, a detection of similarity often becomes problematic [7]. At the root of this problem is that for efficiency reasons search programs give only the longest common connected regions of the

query and a database entry. The end result is that even random identities may "mask" the biologically significant patterns if they are quite short [5,6] and also fragmented matches can remain undetected. Special tools have been developed to overcome this problem. Many of them rely on the notion of domains which are typically short protein-subsequences identified as crucial for determining either the proteins' 3D structure or the biological function, or they have been identified by human inspection as typical repetitive building blocks of a large number of proteins. Domains are grouped in domain-types which are defined in such a way that knowing the domain-type of a protein (i.e. whether at least one of the domains in the type is a subsequence of the given protein) provides a large amount of knowledge about the biological function of that protein. So an interesting and fundamental question is the determination of the domain-types of a protein. Unfortunately, this is again a hard task which cannot be solved by string-matching partly because the domains are too short and thus random identities may disturb the result of string-matching.

In one main track of research a given set of domains is subject to multiple alignments depend on the investigated protein group, which in turn are translated into a compact "consensus representation" of the group. Such consensus representations can take the form of regular expressions [4], frequency matrices (in which the frequency of occurrence of each amino acid in each position of the multiple alignments is represented) [10] or Markov chains [15]. In each case, a "core set" of domains has to be collected to build up the multiple alignments. Whichever way this core set is selected, the resulting consensus representation will be biased. Another more serious problem comes from the fact that the construction of alignments requires human intervention so the basic problem of data explosion cannot be overcome in this way.

Previously we have advocated another strategy in which domain collections were used in conjunction with database searches [9,14]. While this strategy does not need a consensus representation, the selection of domain sequences is however only partly automated and the evaluation of results still requires needs human knowledge and experience.

The aim of the present work is to utilize the capabilities of artificial neural networks in conjunction with database searches to fully automate the recognition of domain-types of proteins. Neural networks have been used in various fields of computational biology, but this is the first application that we are aware of that is devoted to domain recognition.

To introduce our method let us pick out a domain type. The idea is that a corresponding set of representative domains is selected based on a simple statistics. When an unclassified protein enters the system the protein is matched against each of the representative domains, the resulting string-matching numbers being renormalized and used as the input of a feedforward network. The net is trained by a variant of the backpropagation algorithm with the desired output corresponding to whether the investigated protein is in the domain-type or not. An integral part of our strategy is to include string-matching as the first step, in the same way as biologists would evaluate any new sequence. The neural net is used

to combine the string-matching values in a non-linear fashion, which suppresses the effect of accidental and may strengthen weak but biologically important ones. We evaluate the results using a cross-validation test and a confidence interval of the prediction is estimated using on Hoeffding's inequality [11].

2 Method

The method consists of several steps:

1. Suitable preparation of the data
2. Selection of representative domains and preprocessing
3. Training of networks and performance evaluation

2.1 Preparation of data

On the experiments the SWISS-PROT-33 [3] protein and SBASE-5.0 [14,9] domain databases were used. Proteins of SWISS-PROT-33 are categorised as belonging or not belonging to the domain type and sometimes as ambiguous. SBASE was employed to resolve name-standardization problems. Moreover the domains defining the domain type were collected from SBASE [14].

2.2 Selection of representative domains and preprocessing

Since the number of defining domains is usually large ($100 - 400$) a representative subset of the defining domains was selected which would be used in the subsequent processing steps. The representative domains were selected based on their average threshold (string-matching based) similarities to the proteins which belong to the given domain-type, i.e. domains which were very similar to a large number of proteins formed in the representative set. If two domains had the same average threshold similarity then the one with the largest average (unthresholded) similarity value was preferred. This rule was actually needed in a number of cases to decide between two domains. The number of representation domains k was in the range $2 - 9$ and its final value was determined empirically.

Typically the number of proteins belonging to a domain-type is very small compared to the number of all proteins in the database. This calls for a preprocessing step when proteins which are probably not domain-type members are dropped out. The basis of this decision was that if the protein is not similar "enough" to at least the 70 % of the representative domains then the protein could not be a domain-type member. A large number of proteins were eliminated in this way. The error rate of this rule is practically zero, i.e. this rule is very conservative. In some cases this rule still filters out a substantial number of proteins. In this way the neural net could concentrate on learning the characteristics of the more intriguing cases.

The input of the network is a real-vector of k elements, the i th entry of which is the normalized deviation of two similarities. The first is the similarity of the protein to the i th domain and the second is the average similarity of

the proteins within the domain-type. In this way the i th input is positive if the protein is more similar to the i th domain than an “average” protein in the group. The normalization factor is the average similarity, too. There is one exception to the above however: if the protein’s similarity to the i th domain is below threshold then the corresponding input is set to -10 , meaning that the protein is completely dissimilar to the i th domain.¹

2.3 Training and performance evaluation

Several network architectures (including the number of hidden layers and the number of input vector elements) were trained and tested. The MBP package of D. Anguita [2] which implements a batch backpropagation algorithm with an adaptive learning rate and momentum [16] was employed. It was consistently observed that increasing the number of input vector elements helped when this number was small, but not for larger numbers. Using Occam’s Razor we present the results for the smallest nets which achieved the near best performance. The training and test set were separate (i. e. we used a cross-validation test) and Hoeffding’s inequality was used to give a confidence interval for the estimation error. If $L_n = P(g_n(X) \neq Y | D_n)$ denotes the (true unknown) probability of error of the classifier g_n when trained on data D_n ($D_n = ((X_1, Y_1), \dots, (X_n, Y_n))$ where $(X_i, Y_i), (X, Y), i = 1, 2, \dots, n$ are i.i.d. random variables) and this error is estimated using a separate test set $T_m = ((X_{n+1}, Y_{n+1}), \dots, (X_{n+m}, Y_{n+m}))$ as the relative frequency of the error made by g_n on T_m , i.e. by $\hat{L}_{n,m} = (1/m) \sum_{j=1}^m I_{\{g_n(X_{n+j}) \neq Y_{n+j}\}}$ then the distribution of $m\hat{L}_{n,m}$ is binomial with parameters m and L_n , so by Hoeffding’s inequality [11] (see also Corollary 8.1 of [8]) for any given $\epsilon > 0$

$$P(|\hat{L}_{n,m} - L_n| > \epsilon | D_n) \leq 2e^{-2m\epsilon^2}.$$

From this inequality a confidence interval of size ϵ can be estimated at any given error-probability rate (significance level).

3 Results

As a test set we chose a number of known domain types, in graded order of difficulty, to recognize the KRINGLE, ANK, EGF, and FN3 domain types [4]. The KRINGLE domain-type can be identified more or less precisely with regular expression type searches. The ANK domain-type is more problematic, having no characteristic sequence signature. The EGF domain-type contains many variants (these are termed EGF-LIKE in the SWISS-PROT-33 database) and this makes the recognition of the corresponding proteins harder. Finally, the FN3 is an especially variable group, with only 3 constant amino acids within a segment of about 100. Therefore the traditional methods perform very poorly on

¹ Since the great majority of the input vector component values lay in the interval $[-1, 1]$ interval we did not experience any saturation problems so that preprocessing could be kept simple.

this domain-type. Our results are presented in Table 1. The available data was divided into 2:1; that of the data and test set. The number of elements of the positive/negative samples and total number of samples for the four domains are shown in the table. The table also highlights the respective number of mistakes made by the neural net, the mean error estimate (i.e. the errors of the first and second kind), the 0.95-confidence radius and 95% upper bound for the error of the net. Note that the confidence-interval estimates are very conservative and reflecting that using only a small number of samples it is hard (if not impossible) to estimate the actual error-probability. Nevertheless the number of errors was usually very small, i.e. the estimated error-probability ranged from 0 to 0.08. According to our experiences the number of errors increases from KRINGLE to FN3. We would like to notes that we have identified a number of proteins which in the higher version of SWISS-PROT-33 (i.e. in SWISS-PROT-34) have been classified in a way different from the earlier version (which we used for training and testing due to technical reasons) and this now annotations were found to be consistent with neural network prediction!

95%		#proteins	#errors	$\widehat{I}_{n,m}$	ϵ	$\widehat{I}_{n,m} + \epsilon$
	pos.	12	0	0	0.3920	0.3920
KRINGLE	neg.	1932	0	0	0.0308	0.0308
2 : 0	all	1944	0	0	0.0308	0.0308
	pos.	32	0	0	0.2400	0.2400
ANK	neg.	717	9	0.0125	0.0507	0.0632
3 : 3	all	749	9	0.0120	0.0496	0.0616
	pos.	101	4	0.0396	0.1351	0.1747
EGF	neg.	919	20	0.0217	0.0447	0.0664
7 : 7	all	1020	24	0.0235	0.0425	0.0660
	pos.	59	5	0.0847	0.1768	0.2615
FN3	neg.	15636	891	0.0569	0.0108	0.0677
7 : 7	all	15695	896	0.0570	0.0108	0.0678

Table 1. Error-probability Estimates

The first column shows the domain-types to be recognised and size of the neural net in the style (no. of input nodes:no. of hidden layer units). The results are presented in three rows for each domain type: the positive, negative and all examples. This labels are given in column two while in the third and fourth columns the number of corresponding test examples and mistakes on those test sets are shown. Next follows the error-probability estimate, the confidence interval radius at the 95% confidence level and the 95% pessimistic estimate of the error-probability.

4 Conclusions

A neural-network based solution for the fully automatized recognition of domain-types of proteins has been set fasth by us. The network having been tried out

on four different domain types of different recognition-hardness. We found that the neural nets performed very well even for hard-to-recognize domain-types like FN3. As far as is known our method is the first to evaluate a complete protein database. One of the reasons for this is probably because our method is an order of magnitude faster than other methods like PROFILE-SEARCH [4] and PFAM [15]. In the future we plan to test our method for all the domain-types (these are over 1000 such domain-types) which are included in SBASE-5.0. The final goal is to train a neural-net for each domain-type which can then be used to categorize proteins.

References

1. S.F. Altschul and D.J. Lipman. Protein database searches for multiple alignments. *Proc. Natl. Acad. Sci. USA*, 87:5509–5513, 1990.
2. D. Anguita, G. Parodi, and R. Zunino. An efficient implementation of bp on risc-based workstations. *Neurocomputing*, 1993.
3. A. Bairoch and R. Apweiler. The swiss-prot protein sequence data bank and its new supplement trembl. *Nucleic Acids Res.*, 24:21–25, 1996.
4. A. Bairoch, P. Bucher, and K. Hoffmann. The prosite database, its status in 1995. *Nucleic Acids Res.*, 24:189–196, 1996.
5. W.C. Barker, L.T. Hunt, and D.G. George. *Protein Seq. Data Anal.*, 1:363–373, 1988.
6. M. Baron, D.G. Norman, and I.D. Campbell. Protein modules. *Trends in Biochemical Sciences*, 16:13–17, 1991.
7. P. Bork. *Curr. Opin. Struct. Biol.*, 2:413–421, 1992.
8. L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.
9. P. Fabian, J. Murvai, Zs. Hatsagi, K. Vlahovicek, H. Hegyi, and S. Pongor. The sbase protein domain library, release 5.0: a collection of annotated protein sequence segments. *Nucleic Acids Res.*, 25:240–243, 1997.
10. M. Gribskov, A.D. McLachlan, and D. Eisenberg. Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci.*, 84:4355–4358, 1987.
11. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.
12. D.J. Lipman and W.R. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1436–1441, 1985.
13. W. R. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, 85:2444–2448, 1988.
14. S. Pongor, V. Skerl, M. Cserzo, Z. Hatsagi, G. Simon, and V. Bevilacqua. The sbase domain library: A collection of annotated protein sequence segments. *Protein Engineering*, 6:391–395, 1992.
15. E.L.L. Sonnhammer, Sean R. Eddy, and Richard Durbin. Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins in press*, 1997.
16. T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, and D.L. Alkon. Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, 59:257–263, 1993.
17. W.J. Wilbur and D.J. Lipman. Rapid similarity searches of nucleic acid and protein data banks. *Proc. Natl. Acad. Sci. USA*, 80:726–730, 1983.