Chapter

Towards a memory-based interpretation of proteome data

JÁNOS MURVAI, KRISTIAN VLAHOVIČEK and SÁNDOR PONGOR
International Center for Genetic Engineering and Biotechnology
Padriciano 99, 34012 Trieste, Italy, pongor@icgeb.trieste.it

1. INTRODUCTION

Understanding and managing genomic data has become a major bottleneck of biomedical research that calls for novel informatics approaches. The task is immensely complex: to "understand" the role of a protein for example implies inserting it into a host of interconnected and evolving frameworks of biological knowledge, including 3-D structures, molecular interactions, biochemical pathyways, genomic locations, spatial and temporal roles within the cell, the organism, the population and the species. *Similarity based predictions* play an important role in this process: similar biological functions or roles are mostly inferred from similar structure or similar molecular interactions, etc. This is usually carried out by comparing a protein sequence with a database of known sequences, using such programs as BLAST [1, 2].

The ultimate goal of protein sequence comparison is to find biologically significant sequence similarities that can help one to infer the fold and/or the function of an unknown protein based on its sequence. The fundamental problem is that many of the biologically important sequence similarities are not significant in the statistical sense, i.e. the alignment scores or alignment patterns found between sequences similar in the biological sense can not be well distinguished from chance similarities. This is true both for protein

domains that are shared by different protein superfamilies, as well as for complete sequences of proteins that carry the same function in different organisms (**Figure 1**) .

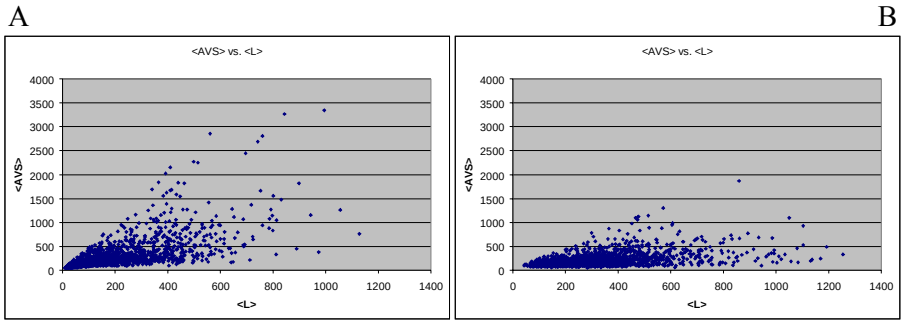

Figure 1 Self-smilarity of various sequence groups as determined by the BLAST program for A) protein domains (SBASE7.0, [15]) and orthologous sequences taken from complete genomes (COG [5]). Each sequence was compared with members of its own group using the BLAST program, and the average similarity score was calculated.

The intuitive notion of "biologicaly significant similarities" can be best described as membership in a similarity group characterized by established similarities, such as a group of domain sequences, or a group of orthologous proteins. For example, the similarity of a, say, trypsin-like domain to other trypsin-like domains is considered as "biologically significant", while its similarities to unrelated sequences are referred to as "chance". Prediction of domain homologies or of protein function is thus different from simple sequence similarity search in the sense that it is not only concerned with finding the most similar sequence to a query, but rather with assigning the query to a known similarity group. Also, statistical significance is allways understood in a context (a database, a scoring system and a theoretical model of chance similarities) but membership in a biologically significant similarity group (e.g. "trypsin-like") is generally used as permanent quality. A great deal of work has been invested into the modeling of sequence similarity groups in terms of Hidden Markov Models, sequence profiles and related techniques [3]. All these approaches provide *individual* descriptions to each group under study and consequently they are difficult to update with the stream of genomic data.

2. REPRESENTATION OF SIMILARITIES

The aim of the present work is to characterize the distribution of similarity scores within the known sequence similarity groups and to

develop a general, exemplar-based prediction strategy on this basis. The examples shown in **Figure 2** in fact suggest that the "self similarity scores" found between the members of various similarity groups (gray circles) are more or less separated from the chance or "non-self" similarities i.e. those that unrelated sequences have with members of the same group (black triangles).

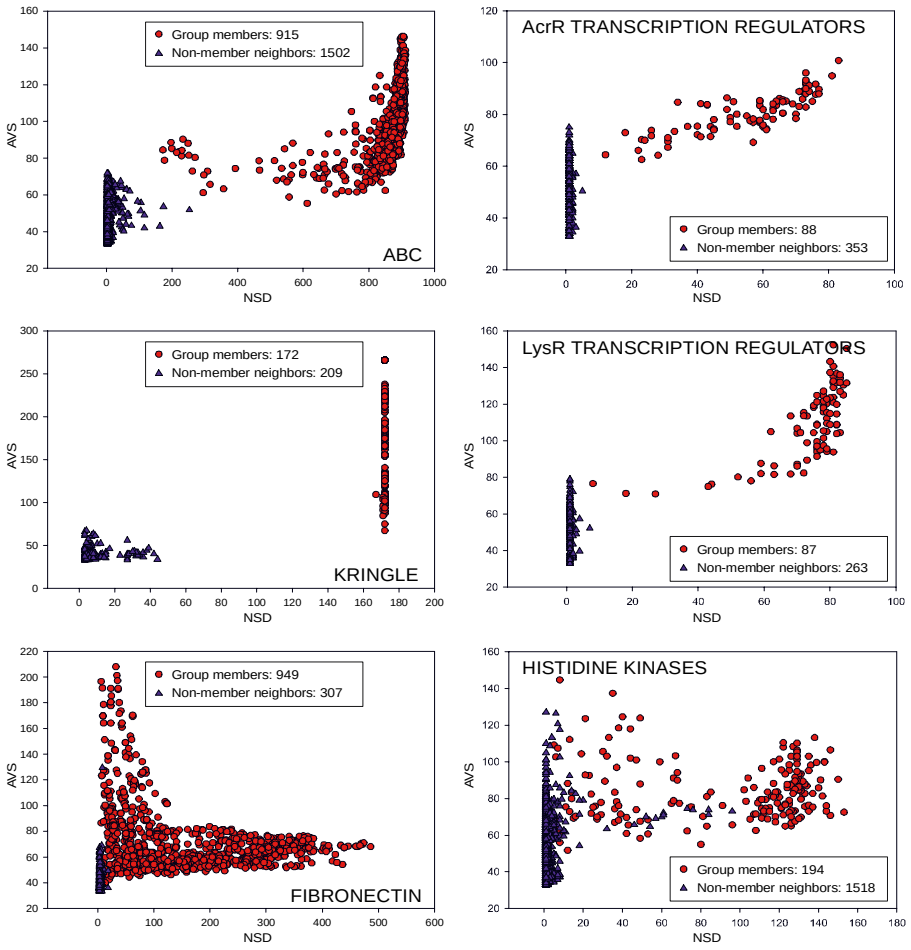

Figure 2 Number of significant similarities (NSD) vs. Average similarity score (AVS) plots in various domain sequence (SBASE) and orthologous protein sequence (COG) groups (Prosite, PFAM or COG reference given in parenthesis): ABC Transporter [PROSITE:PS00211]; Kringle [PROSITE:PDOC00020]; Fibronectin III domain [PFAM:PF00041]; Transcriptional regulators, AcrR family [COG:COG1309]; Transcriptional regulators, LysR family [COG:COG0583]; Sensory transduction histidine kinases [COG:COG0642]. The data are determined from a database vs. database comparison using BLAST. Each dot represents an SBASE domain or a COG sequence. The AVS and NSD values are computed from the BLAST similarities between the sequence and other members

of the group ("self similarities", circles), or sequence and members of other groups ("non-self similarities", triangles)

Some groups are "tight" in the sense that most members are significantly similar to most other members, and, also there are differences in how well the self and non-self similarities are separated from each other. The intuitive concepts of "tightness" and "separation" can be visualized by plotting the empirical distributions the similarity scores as shown in **Figure 3**.

$$C = TP_{NSD} + P_{p}^{NSD}(NSD) + P_{nfp}^{NSD}(NSD) + TP_{AVS} + P_{p}^{AVS}(AVS) + P_{nfp}^{AVS}(AVS)$$

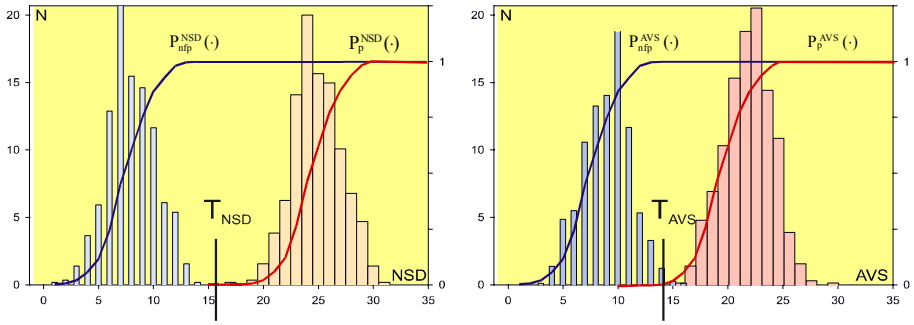

Figure 3 Variables used to characterize self- and non-self similarities. (Schematic histograms representation of the data shown in **Figure 2**). **A)** Empirical distribution of NSD of self-similarity scores $P_{p}^{NSD}(\cdot)$ and non-self similarity scores $P_{nfp}^{NSD}(\cdot)$ **B)** Empirical distribution of AVS of self-similarity scores $P_{p}^{AVS}(\cdot)$ and non-self similarity scores $P_{nfp}^{AVS}(\cdot)$ The four distribution curves are computed from a database vs. database comparison and stored for each sequence group. During prediction, a sequence is compared to the domain (or COG) database with BLAST, and NSD and AVS are computed for all categories present in the output. Then, the values of $P_{p}^{NSD}(NSD), P_{p}^{AVS}(AVS), P_{nfp}^{NSD}(NSD)$, and $P_{nfp}^{AVS}(AVS)$, are simply read out from the precomputed empirical distributions. The cumulative score C is the sum of these variables, plus two threshold premium values TP_{NSD} and TP_{AVS} that are have a value of 2.0 if *NSD* and AVS are above the group threshold T_{NSD} and T_{AVS}; respectively, and zero otherwise. [4].

The approach presented here consists in characterizing the self- and non-self similarity distributions in a reference database, and then predicting group membership by comparing a query to the reference database. In this work we show results on domain sequences, using the SBASE domain sequence database [4] and functional similaritiews, using the COG database of orthologous clusters [5]. The similarity scores are calculared with the BLAST program [1]. The key step of this approach is a simple transformation that represents a sequence object as a vector of similarity scores calculated with respect to a reference sequence database. This is equivalent of placing the sequence object into a similarity space with as

many dimensions as there are members in the reference database. In this space, the object is no longer described in terms of its structure or sequence but rather by its similarities to the database objects. As the reference sequence databases can contain more than 100 thousand sequences, the similarity vectors would, in principle, have a very large number of components, even though most of them negligibly small. A solution to this problem is to filter the similarities by mathematical significance first, (with the BLAST program we achieve this by choosing a generously loose cutoff of P=0.8, ~ E=200), then to calculate group (class) averages of the similarity scores. The similarity of a sequence to a group *g* is then represented by two variables, the number of significant similarities, NSD_g, and the average of the corresponding similarity scores, AVS_g. This is equivalent of dividing the similarity space into small neighborhoods corresponding to each similarity group, and define a class-specific similarity function between the query and each of the sequence groups. Each neighborhood will center around one similarity group, and will contain members of the group as well as well as non-members that nevertheless display significant similarity to various members of the groups, as shown for several domain and COG groups in **Figure 2**. (a complete set of 2-D plots is available at www.icgeb.trieste.it). The variables used to describe each neighborhood are those shown in **Figure 3**. These are local variables in each neighborhood and characterize self-and the non-self similarity within the neighborhood (NSD_g, AVS_g). They are determined from a database vs. database comparison and stored for each neighborhood as a knowledge base of similarities.

3. CATEGORIZATION ALGORITHMS

The categorization is then carried out by first comparing a query sequence with the reference database, then evaluating the search output in comparison with this knowledge base using one of the 3 simple classification methods described below. The interesting point is that after this transformation, many of the most difficult sequence categorization tasks (i.e. those cases where the within-group similarity is is very far from what is considered significant in the statistical sense) can be quite efficiently solved using standard, simple non-parametric classification methods, of which we give three examples.

i) The first such approach is a nearest neighbor algorithm applied to the self similarities: The categories are ranked according the NSD and AVS_g values, and if the winner of both list is same, that category will be assigned to the query, other cases will be reported as "doubt". It is required, however,

that the AVS_g and NSD_g values should be beyond threshold values defined as the smallest respective value found in the given group of the reference database (**Figure 3**). This nearest neighbor method provides correct predictions (no false positives or false negatives) in 80 % of the domain groups and 71 % of the functional groups. Examples are shown in **Table 1**.

ii) A more efficient yet simple scoring method incorporates both self and non-self similarities The cumulative probabilistic score C_g described in **Figure 3** [4] does not require that either of the NSD_g or AVS_g values be beyond the respective thresholds; it will be calculated for all categories present in the search output and the category with the highest C_g value will be assigned to the query. It is used as a simple diagnostic tool, for a sequence to be considered a member of the group, its C score should reach the lowest value recorded for the group - other cases are reported as doubt. **Table 1** shows that the probabilistic score allows correct prediction in many further sequence groups, leaving incorrect predictions in only 112 domain groups (10%) and 194 functional groups (9 %).

iii) For the categories that can not be efficiently handled by either of the two previous methods we may apply more sophisticated techniques. We chose a standard feed-forward artificial neural network (ANN) architecture with 6 hidden units (**Figure 4**), using the 6 parameters shown in **Figure 3** as the input [6]. The ANN method produces correct predictions in 55 domain groups and very low number of errors in the others (**Table 1**). At this stage, however, some of the weaknesses of the approach become apparent: artificial neural networks require a large training set, so they can not be applied efficiently to small domain sequence groups. This problem is especially conspicuous with the COG functional groups some of which are typically quite small. As a workaround, we selected homologous sequences from other databases (Swiss-Prot and PIR), but this is in some cases quite difficult because of the annotation conflicts that exist between the various databases. We developed ANNs only to 5 of the "problematic" functional sequence groups, and the results are as satisfactory as those obtained with the domain groups. While the parameters of the previous two methods are automatically derived from on a database vs. database comparison statistics, the neural network method requires an individual training for each of the sequence groups studied. The architecture

Sequence group					Prediction method						Final performance	
Name	N[1,2]	NS[3]	AVS[4]	E[5]	Nearest neighbor		Probabilistic scoring		Artificial neural network			
					correct	fp/fn	correct	fp/fn	correct	fp/fn	correct	fp/fn
SBASE release 7.0												
KRINGLE DOMAIN	61	172.92 (172-173)	156 (69.69-273.01)	1.27E-14	61	0/0	61	0/0	-	-	61	0/0
COLLAGEN TRIPLE HELIX REPEAT	211	879(878-879)	144.67(89.52-181.48)	3.82E-13	211	4/0	211	0/0	-	-	211	0/0
7 TRANSMEMBRANE RECEPTOR	825	650.84(23-798)	115.75(56.84-274.17)	2.10E-08	825	6/0	825	0/0	-	-	825	0/0
CYTOCHROME C OXIDASE SUBUNIT II.	233	294.66(17-406)	290.62(67-409.91	1.45E-33	233	1/0	233	0/0	-	-	233	0/0
ABC TRANSPORTERS	769	54.83 (6-104)	196.07 (74.62-440.93)	6.39E-20	769	11/0	769	6/0	769	1/0	769	1/0
SUSHI DOMAIN (SCR REPEAT)	165	287.10 (19-446)	73.12 (4407-155.45)	4.92E-03	165	30/0	165	3/0	165	0/0	165	0/0
PROTEIN-KINASE DOMAIN	1642	1456.05 (13-1621)	112.41 (54.36-273.00)	6.28E-08	1640	9/2	1642	30/0	1642	0/0	1642	0/0
TRYPSIN-LIKE DOMAIN	442	380.94 (9-419)	125.75 (47.94-353.36)	6.86E-10	442	170/0	442	3/0	442	0/0	442	0/0
ANK REPEAT	157	193.32 (5-447)	61.93 (46.43-90.24)	1.09E-01	156	33/1	157	4/0	155	2/2	155	2/2
FIBRONECTIN TYPE III DOMAIN	350	146.86 (5-489)	93.40 (54.05-267.33)	9.99E-06	350	514/0	350	8/0	350	2/0	350	2/0
EGF-LIKE DOMAIN	436	741.08 (3-1271)	53.86 (35-73.53)	1.60E+00	433	181/3	436	34/0	436	3/0	436	3/0
IMMUNOGLOBULIN DOMAIN	2584	1040.36 (4-1791)	120.66 (41.90-345.63)	1.20E-09	2584	3011/0	2584	633/0	2552	71/32	2552	71/32
Of a total of 52933 sequences					45117	7592/250	5775	2041/0	1165	166/99	52389	445/99
Of a total of 1515 of groups					1213	270/112	154	147/0	113	55/54	1427	90/54
COG												
Transcriptional regulators, LysR family	87	73.63 (8-85)	109.47 (70.92-152.38)	2.41E-07	87	0/0	87	0/0	-	-	87	0/0
Acyl-CoA dehydrogenases	58	48.20 (18-57)	139.07 (64.39-223.82)	2.72E-11	58	0/0	58	0/0	-	-	58	0/0
Transcriptional regulators, AcrR family	88	53.59 (12-83)	80.54 (62.56-100.81)	1.79E-03	86	0/2	88	0/0	-	-	88	0/0
Dipeptide/oligopeptide/nickel ABC-type transport systems, periplasmic components	62	48.25 (8-60)	112.80 (64.66-217.37)	1.55E-07	59	0/3	62	0/0	-	-	62	0/0
Glycosyltransferases I	107	48.20 (2-90)	86.27 (51.0-388.89)	5.51E-04	89	0/18	107	1/0	300	2/0	300	2/0
Sensory transduction histidine kinases	195	89.93 (1-153)	83.75 (46.0-144.62)	1.14E-03	145	1/49	194	6/1	264	2/4	264	2/4
Thiol-disulfide isomerase and thioredoxins	104	34.63 (1-72)	89.17 (48.0-156.0)	9.81E-05	79	0/25	104	4/0	218	1/6	218	1/6
Serine/threonine protein kinases	154	116.48 (4-146)	98.96 (59.88-147.58)	1.35E-05	130	0/22	152	2/2	1622	1/12	1622	1/12
Permeases	363	95.12 (1-238)	105.07 (38.5-447.17)	1.45E-06	288	0/75	363	0/5	428	29/16	428	29/16
Of a total of 28141 sequences					24728	71/3429	27457	684/684	888	35/35	27457	684/684
Of a total of 2104 of groups					1494	48/658	1910	194/194	5	5/4	1910	194/193

[1] The comparisons were done with the BLAST program.
[2] Number of sequences in the group
[3] Number and range (max-min) of significant similarities to members of the same group
[4] Average similarity score and range (max-min) of significant similarities to members of the same group
[5] Expect value, calculated as $E = mn \cdot 2^{-S}$; $S = \frac{\lambda \cdot AVS - \ln K}{\ln 2}$, where m= 31411157, n= 100, k=0.138, lambda=0.318

is general, however, i.e. it does not depend on the domain group so the the training is automated. In addition, the trainig process takes a very short time (on the average it took less than 2 minutes for a domain group, using a 300 MHz single processor Sun work-station).

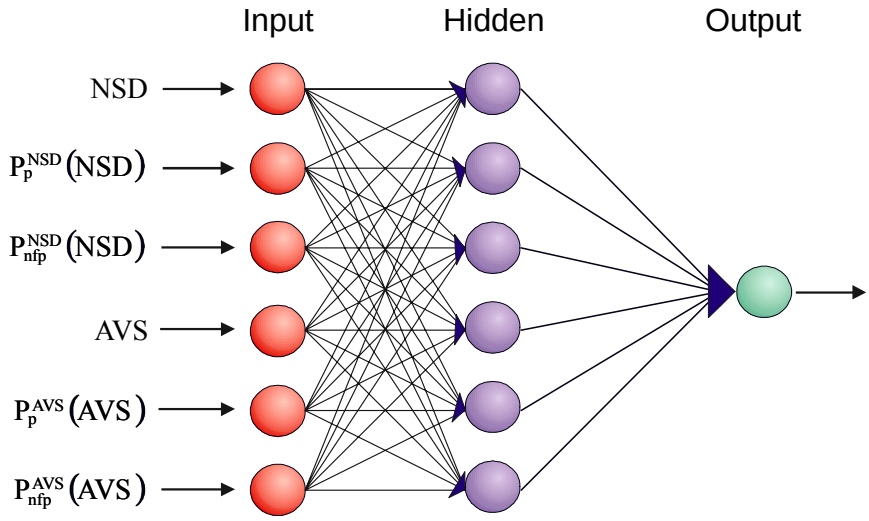

Figure 4 The backpropagation neural network architecture used for domain recognition. The variables are explained in the text and in **Figure 3**.

WWW servers for domain and function prediction are available at www.icgeb.trieste.it/sbase

4. DISCUSSION

The predictive performance of these methods compares quite favorably with other methods such as Hidden Markov models [7] and profile searches [8]. Essentially, the present approach provides efficient categorization in the time frame of a simple sequence similarity search, and does not even require iterative re/searching of the database. We believe that the success of this approach is primarily due to the similarity-based encoding of the data. This is apparent from the fact that artificial neural networks with only 6 input data could efficiently handle the categorization in all cases in which there was a sufficient number of data available for training.

We term the present method a memory-based approach, because its principles, especially the nearest neighbor algorithms *i* and *ii* are analogous in many respect to the memory-based learning paradigm described by Stanfill and Waltz [9].Clearly, NSD, AVS and the probabilistic score C can be regarded as class-specific similarity (distance) functions that have parameters (thresholds, frequency distributions) which are learned from the database. Second, the reference database and the similarity knowledge base can in fact be considered as the memory of the system. This memory consists of two parts, a) Information expressed in terms of propositions added by human annotators (class labels, i.e. definition of domains or functions), and b) Information stored in inter-object connections that are determined automatically for each group from database vs. database comparisons (class average similarities, thresholds and and empirical distributions). It is noted that the simple, memory-based categorization strategies i) and ii) proved successful in the vast majority of the cases, neural networks had to be used for some of the very difficult and large groups. It is worth mentioning that similarity groups with low average statistical significance and short in sequence required more sophisticated methods (ii) and iii), the others could be handled by the simple, common sense nearest neighbor algorithm. It is important to point out the cases that could not be properly categorized by this approach: Out of 1515 domain groups, 17 such groups were found; they were not sufficiently separated for algorithms i and ii, but had too few data for training ANNs. On the other hand, out of 2104 COG clusters only 194 could not be categorized without mistakes using the simple algorithms; in those cases, however, annotation conflicts (between COG, Swiss-Prot and PIR) may have been the underlying reason. This points to the known fact that nearest neighbor type methods are quite sensitive to database errors [10].

The advantage of the current method is its speed and simplicity. Sequence classification tasks are typically solved by strategies that are based on individual models of the sequence groups described in terms of. HMMs and profiles etc. These are computationally intensive strategies (as they require multiple alignments) and require a substantial human overhead for updating. The current method offers a comparable if not better performance (**Table 1**) moreover it is quite simple to update. Adding a new sequence group to the reference database consists in comparing the members of the new group to each other and to the database using BLAST, defining the 6 class specific quantities shown in Figure 3 (The two thresholds and the 4 P distribution curves) and updating the corresponding data of those classes that showed significant similarities with the new group. The speed of the method is a consequence of the encoding and data reduction schemes

developed on the basis of earlier observations [11, 12]. Grouping and evaluating the scores by classes means that only the immediate neighborhood of the similarity groups, i.e. a small fraction of the entire similarity space is explicitly represented, which however contains practically all detectable similarities (i.e. those with significance values $P<0.8$). Typically, a domain sequence will have significant similarities to 4 goups on the average, so the number of cases to be tested is not too large.

The present approach is conceptually different from most other sequence classification methods, because it uses an exemplar-based description [13] of the similarity groups. The classification methods used by the COG [5] and the SYSTERS [14] databases are also exemplar based, while most other methods use consensus-models that are probabilistic descriptions of the similarity groups [13]. Exemplar-based methods have the important advantage that they are not statistically biased, i.e. allow the detection of atypical examples in the presence of an excess of known typical examples.

The present approach here is not limited either to the application examples or to the simple algorithms described here. All three algorithms were chosen because of their simplicity and none of them is considered as optimal in any sense; preliminary results show that adaptive kernel methods and support vector machine recognizers may provide useful alternatives [10]. The choice of application areas will basically depend on the availability of a metrics distance (similarity measure) capable of distinguishing the object classes. It is important to note that the object classes used in this work were conceptual constructs based on human knowledge. It is equally possible to use classes automatically generated by clustering algorithms. Future application areas may include e.g. prediction of metabolic pathways, evolutionary networks and related classification schemes.

REFERENCES

1. Altschul, S.F., et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.* Nucleic Acids Res, 1997. **25**(17): p. 3389-402.
2. Altschul, S.F., et al., *Basic local alignment search tool.* J Mol Biol, 1990. **215**(3): p. 403-10.
3. Atwood, T.K., *The role of pattern databases in sequence analysis.* Briefings in Bioinformatics, 2000. **1**(1): p. 45-59.
4. Murvai, J., K. Vlahovicek, and S. Pongor, *A simple probabilistic scoring method for protein domain identification.* Bioinformatics, 2000: p. in press.
5. Tatusov, R.L., et al., *The COG database: a tool for genome-scale analysis of protein functions and evolution.* Nucleic Acids Res, 2000. **28**(1): p. 33-6.

6. Murvai, J., et al., *Prediction of protein functional domains from sequences using artificial neural networks.* Genom Research, 2000: p. in press.
7. Sonnhammer, E.L.L., S.R. Eddy, and R. Durbin. *Pfam: A Comprehensive Database of Protein Domain Families Based on Seed Alignments*. in *Proteins in press*. 1997.
8. Gribskov, M., A.D. McLachlan, and D. Eisenberg, *Profile analysis: Detection of distantly related proteins.* Proc. Natl. Acad. Sci., 1987. **84**: p. 4355--4358.
9. Stanfill, C. and D. Waltz, *Toward memory-based reasoning.* Communications of the ACM, 1986. **29**(12): p. 1213-1228.
10. Ripley, B.D. and N.L. Hjort, *Pattern Recognition and Neural Networks*. 1995, Cambridge: Cambridge university press.
11. Hegyi, H. and S. Pongor, *Predicting potential domain homologies from FASTA search results.* Comput Appl Biosci, 1993. **9**(3): p. 371-2.
12. Murvai, J., et al., *The domain-server: direct prediction of protein domain-homologies from BLAST search.* Bioinformatics, 1999. **15**(4): p. 343-4.
13. Smith, E.E. and D.L. Medin, *Cathegories and concepts*. Cognitive science series 4. 1981, Cambridge, Massachusetts; London, England: Harvard university press.
14. Krause, A., J. Stoye, and M. Vingron, *The SYSTERS protein sequence cluster set.* Nucleic Acids Res, 2000. **28**(1): p. 270-2.
15. Murvai, J., et al., *The SBASE protein domain library, release 7.0: a collection of annotated protein sequence segments.* Nucleic Acids Res, 2000. **28**(1): p. 260-2.