A Multi-Expert System for the Automatic Detection of Protein Domains from Sequence Information

Niranjan Nagarajan and Golan Yona*
Department of Computer Science
Cornell University
* Corresponding author: golan@cs.cornell.edu

ABSTRACT

We describe a novel method for detecting the domain structure of a protein from sequence information alone. The method is based on analyzing multiple sequence alignments that are derived from a database search. Multiple measures are defined to quantify the domain information content of each position along the sequence, and are combined into a single predictor using a neural network. The output is further smoothed and post-processed using a probabilistic model to predict the most likely transition or boundary positions between domains. The method was assessed using the domain definitions in SCOP for proteins of known structures and was compared to several other existing methods. Our method improves significantly over the best method available, the semi-manual PFam domain database, while being fully automatic. Our method can also be used to verify domain partitions based on structural data. Few examples of predicted domain definitions and alternative partitions, as suggested by our method, are also discussed.

Categories & Subject Descriptors: I.5 Pattern Recognition, I.2.6 Learning, H.1.1 Systems and Information Theory, J.3 Life and Medical Sciences—Biology and genetics.
General Terms: Algorithms, Theory.
Keywords: protein domains, domain prediction, SCOP, domain boundaries

1. INTRODUCTION

One of the first steps in analysing proteins is to detect the constituent domains or the **domain structure** of the protein. A domain is considered as the fundamental unit of protein structure, folding, function, evolution and design [1, 2, 3]. It combines several secondary structure elements and motifs, not necessarily contiguous, which are packed in a compact globular structure. It is commonly believed that a domain can fold independently into a stable three dimensional structure and it has a specific function. A protein may be comprised of a single domain or several different domains, or several copies of the same domain. It is the domain structure of a protein that determines its function, the biological pathways in which it is involved and the molecules it interacts with.

Detecting the domain structure of a protein is a challenging problem. Given the protein sequence there are no signals or signs that indicate when one domain ends and another begins. Structural information can help in detecting the domain structure of a protein. Domain delineation based on structure is currently best done manually by experts. The SCOP domain classification [4], which is based on extensive expert knowledge, is an excellent example. However, structural information is available for only a small portion of the protein space. Therefore, there is a strong interest in detecting the domain structure of a protein directly from the sequence.

In our study we define a domain to be a **continuous** sequence that corresponds to an elemental building block of protein folds - a subsequence that is likely to be stable as an independent folding unit. As such we believe that this building block was first formed as an independent protein with a specific acquired function. In the course of evolution, the domain might have been combined with additional domains to perform other, possibly more complex functions. However, if the domain indeed existed at some point as an independent unit, then it is likely that traces of the autonomous unit might exist in other database sequences, possibly in lower organisms. Thus a database search can sometimes provide us with ample information on the domain structure of a protein. For example, the histogram and profile of sequence matches one can obtain from a database search may help to detect domain boundaries [5, 6, 7]. However, one should be cautious in analysing database matches in search for such signals. One possible difficulty arises from the fact that pairs of sequence domains may appear in many related sequences, thus hindering the ability to discern the two apart. Furthermore, mutations, insertions and deletions blur domain boundaries and make it hard to distinguish a signal from background noise.

1.1 Related studies

Previous methods for sequence-based domain detection could be roughly classified into four categories: (i) Methods based on the use of similarity searches and knowledge of sequence termini to delineate domain boundaries using heuristics. Methods like MKDOM [8], Domainer [9], DIVCLUS [10] and DOMO [11] fall in this category. These methods were designed to partition all the proteins in a database into domains but they are in general less accurate due to their heuristic nature. (ii) Methods that rely on expert knowl-

RECOMB'03, April 10–13, 2003, Berlin, Germany.

edge of protein families to construct models like HMMs to identify other members of the family. PFam A [12, 13], TigrFam [14] and SMART [15] fall in this category. These methods are considerably more accurate but are restricted by their ability to make predictions only for well studied families. (iii) Methods that try to infer domain boundaries by using sequence information to predict tertiary structure first. SnapDragon [16] and Rigden's covariance analysis [17] are examples of this approach. These methods use novel sources of information but are computationally expensive. (iv) Methods that use multiple alignments to predict domain boundaries such as PASS [6] and Domination [7]. (v) Other methods, that do not fall into any of the previous categories (clustering sequence alignments [18] and domain guess by size [19]).

There is no fixed, universally accepted set of rules for partitioning a protein into its constituent domains. Therefore it is hard to assess the quality of domain predictions by any of the above algorithms.In the absence of a common framework for analyzing the quality of domain predictions, the various works that we have mentioned above have relied on a variety of qualitative and quantitative evaluation criteria, external resources and manual analysis to verify domain boundaries and study the capabilities of their systems. For example, the quality of domain predictions in DOMO is analyzed by taking domain annotations in PIR [20] and SwissProt [21] as being the standards of truth, and comparing the predictions to ProDom predictions. However, their analysis is based only on a few selected examples.Others, such as Domination and Rigden's covariance analysis, run a more extensive evaluation based on comparisons with structure-based domain definitions as in SCOP [22] but they did not evaluate the capabilities of other methods with this setup.

The diversity of evaluation criteria has made it impossible to objectively compare the various methods for domain prediction. Here we propose and use a common framework to evaluate the various methods. This framework is based on using definitions from the SCOP database as the standard of truth. In addition we devise scores that can be used in a uniform and unbiased fashion to evaluate the accuracy and coverage of the various methods.

Despite the large number of studies, the task of constructing an accurate and efficient general-purpose domain detection system that works solely on sequence information is still an open problem. While methods like SMART and TigrFam are accurate, they require careful manual inspection and provide predictions for a small subset of the sequence database. On the other side of the spectrum, methods like DOMO and ProDom are fully automatic and give predictions for nearly all proteins in the sequence database, but are not sufficiently accurate. In this paper we suggest a novel approach that incorporates many of the salient features of earlier systems into a probabilistic framework that is extensible and is based on rigorous analysis of information sources in order to predict domain boundaries with high accuracy and coverage.

The paper is organized as follows. We first describe the data set, scores and our learning methodology in detail. We then present the results of testing our method on a large collection of proteins with known structures and compare our predictions to structure based domain definitions as well as to other sequence based domain partitioning methods. We conclude with a few examples.

2. METHODS

Given a query sequence, our algorithm starts by searching a large sequence database and generating a multiple alignment of all significant hits. The columns of the multiple alignment are analyzed using a variety of sources, to define scores that reflect the domain-information-content of alignment columns. Information theory based principles are employed to maximize the information content. These scores are then combined using a neural network to label single columns as core-domain or boundary positions with high accuracy. The output of the artificial neural network is then post-processed to smooth and refine predictions while considering local information from multiple columns. Finally, we introduce the domain-generator model that uses global information about the distribution of domain sizes and sequence divergence to test multiple hypotheses, filter out positions that are incorrectly predicted as boundary positions and output the most likely partition. An overview of our method is depicted in Fig. 1. We now turn to describe our method in detail.

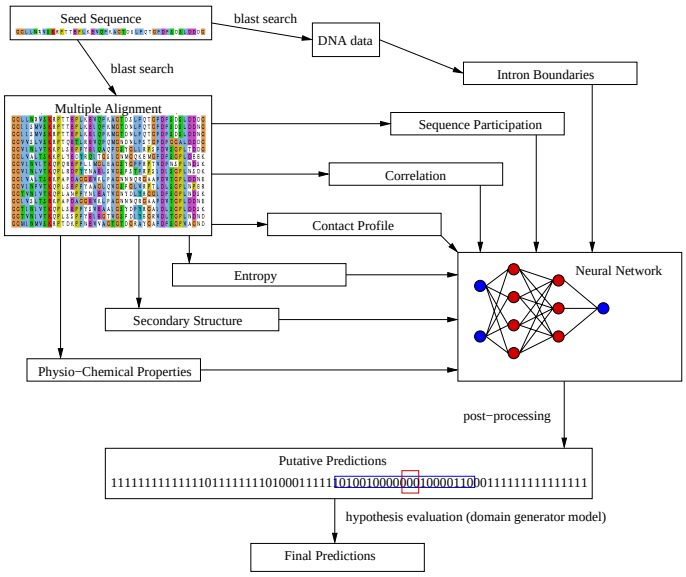

Figure 1: (a) Overview of the domain prediction system.

2.1 The data sets

2.1.1 *The query data set*

In the absence of general rules or principles that define domain boundaries, one must rely on existing knowledge of protein domains, to devise a reliable and accurate methods for automatic domain detection. One of the most extensive collections of protein domains is the one provided by the SCOP classification of protein structures [22]. The domains in this database are defined from PDB records [23].

To train and test our method we selected complete protein chains from PDB, searched the database and generated multiple alignments. About half of these alignments with their corresponding domain structure as defined by SCOP were used for training. The other half was used for testing.

Our initial dataset was the set of protein sequences in the PDB database as of May 2002 with 35,184 protein chains,

and 11,969 non-identical sequence entries. All sequences shorter than 40 amino acids and fragments of longer sequences were eliminated and of all sequences that are more than 95% identical only a single representative was retained, yielding a total of 4,810 valid queries.

2.1.2 Alignments

Each one of the 4810 queries was searched against a composite non-redundant database that contains 933,075 unique sequence entries. The database is composed from 97 different databases among which are SwissProt, TrEMBL, PIR, PDB, SCOP, DBJ, GenBank, REF, PATAA, PRF and the complete genomes of 78 organisms. All entries that are documented as fragments (according to at least one source database) were eliminated, leaving a total of 693,912 non-fragmented entries. The alignment was created in two phases. First, the query was searched against the non-redundant database using BLAST [24] and the related sequences were compiled into a database (a different database for each query sequence). In the second phase, the query was searched against this smaller database, using PSI-BLAST [24] until convergence. Of these alignments, fragmented queries were eliminated and only alignments with more than 20 hits were kept. Finally, the query sequences were grouped into clusters (using the ProtoMap clustering algorithm [25] with a conservative e-value threshold) and from each group only one representative was selected (the one with the maximal number of database aligned sequences). The final set of queries consisted of 3,140 PDB sequences, with their corresponding alignments. Alignments are represented as a sequence of alignment columns with each one being associated with one position in the seed sequence (insertions respective to the seed sequence are processed as described in section 2.2.3).

2.1.3 Domain definitions

The domain definitions are retrieved from the SCOP database, version 1.57 as of May 2002. Of the 3,140 PDB queries, 3,039 were documented in this list, with the number of domains ranging from 1 to 7. In a final pruning step, protein chains that are less than 90% covered by SCOP domains are eliminated. In the final data set we retained all the multi-domain proteins (605) and one-fourth of the single domain proteins (576) to ensure an equal representation of both.

For each protein chain we defined the **domain positions** to be the positions that are at least x residues apart from a domain boundary. Domain boundaries are obtained from SCOP definitions where for a SCOP definition of the form $(start_1, end_1)..(start_n, end_n)$ the domain boundaries are taken to be $(end_i + start_{i+1})/2$. All positions that are within x residues from domain boundaries are considered **boundary positions**. This process allows us to classify all the positions in the proteins being considered as domain or boundary positions.

2.2 The domain-information-content of an alignment column

To quantify the likelihood that a sequence position is part of a domain, or at the boundary of a domain we defined several measures based on the multiple alignment that we believe reflect structural properties of proteins and would therefore be informative of the domain structure of the seed protein.

2.2.1 Conservation measures

Amino acid entropy: Multiple alignments of protein families can expose the core positions along the backbone that are crucial to stabilize the protein structure, or play an important functional role (as in the active site or in an interaction site). These positions tend to be more conserved than others and strongly favor amino acids with similar and very specific physio-chemical properties, because of structural and functional constraints. One possible measure of the conservation of an alignment column is given by the entropy of the corresponding distribution. For a given probability distribution $\mathbf{P}$ over the set $\mathbf{A}$ of the 20 amino acids $\mathbf{P} = (p_1, p_2, ..., p_{20})^t$, the entropy is defined as

$$E_a(\mathbf{P}) = -\sum_{i=1}^{20} p_i \log_2 p_i$$

For a given alignment column, the probability distribution $\mathbf{P}$ is defined from the empirical counts, after adding pseudo counts as described in [26].

Class entropy: Quite frequently one may observe positions in protein families that have a preference to a **class** of amino acids, all of which have similar physio-chemical properties. The amino acid entropy measure is not effective in such cases since it ignores amino acid similarities. An entropy measure based on suitably defined classes may capture positions with subtle preferences towards classes of amino acids. The classes we use are sulphur (CM), simple aliphatic (AL), side-chain restrictive aliphatic (IV), aromatic (FWY), hydroxyl (ST), amide (NQ), acidic (ED), Basic (KRH), proline (P) and glycine (G). This classification (Linda Nicholson, personal communication) worked better than other classifications that we found in the literature [27, 28].

Given the set $\mathbf{C}$ of amino acid classes and the empirical probabilities (with pseudo counts) $\mathbf{P}$ the class entropy is defined in a similar way to the amino acid entropy

$$E_c(\mathbf{P}) = -\sum_{i \in C} p_i \log_2 p_i$$

Evolutionary pressure: The class entropy is one possible solution to the aforementioned problem, however, it does not utilize all the prior information we have about amino acid similarities. A better entropy measure would consider the mutual information (similarity) of the amino acids. To the best of our knowledge, this problem has never been addressed directly before. A possible extension may generalize upon the results of [29]. Alternatively, we suggest a measure that estimates the evolutionary pressure in an alignment column by calculating the evolutionary span, approximated by the sum of pairwise similarities of amino acids in a column. Specifically, if the number of sequences "participating" in an alignment column k is n then the span of this column is defined as

$$Span(k) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j<i} s(a_{ik}, a_{jk})$$

where a_{ik} is the amino acid in position k of sequence i and $s(a, b)$ is the similarity score of amino acids a and b according to a scoring matrix such as BLOSUM50 [30].

2.2.2 Consistency and correlation measures

Since protein domains are believed to be stable building blocks of protein folds, it is reasonable to assume that all appearances of a domain in database sequences will maintain the domain's integrity. Integrating the information from multiple sequences can generate a strong signal, indicative of domain boundaries by detecting changes in sequence participation and evolutionary divergence. Several different measures are tested. These measures quantify the correlation and consistency of neighboring columns in an alignment.

Consistency: This simple coarse-grained measure is based on sequence counts. The measure is defined as the difference in sequence counts of a column and the average of the surrounding columns in a window of size w. If c_k is the sequence count in position k then

$$Consistency(k) = |c_k - \frac{1}{2w} \sum_{i \neq k, |i-k| \leq w} c_i|$$

Asymmetric correlation: This is a more refined measure that considers the consistency of individual sequences and sums their contributions. To measure the correlation of two columns we first transform each alignment column into a binary vector of dimension n (the number of sequences in the alignment) with 1's signifying aligned residues and 0's for gaps. Given two binary vectors $\vec{u}$ and $\vec{v}$ their asymmetric correlation (bitwise AND) is defined as

$$Corr_a(\vec{u}, \vec{v}) = < \vec{u}, \vec{v} > = \sum_{i=1}^{n} u_i \cdot v_i$$

High correlation values reflect consistent sequence participation while low correlation values signal a region of ambiguous sequence participation and possible domain boundaries.

Symmetric correlation: the asymmetric correlation does not reward for sequences that are missing from both positions. However, these may reinforce a weak signal based only on participating sequences. The symmetric correlation corrects this by using bitwise XNOR when comparing two alignment columns, i.e.

$$Corr_s(\vec{u}, \vec{v}) = \sum_{i=1}^{n} \delta(u_i, v_i)$$

where δ is the delta function $\delta(x, y) = 1 \iff x = y$

To enhance the signal and smooth random fluctuations the contributions of all positions in a local neighborhood around a sequence position are added, and all correlation measures for an alignment column are calculated as the average correlation over a window of size w centered at the column (the parameter w is optimized, as described in section 2.4).

Sequence termination: sequence termination is a strong signal of a domain boundary. However, in a multiple alignment it is not necessarily indicative of a true sequence termination. Although we eliminated all sequences that are documented as fragments from our database, the sequence may still be a fragment of a longer sequence without being documented as such. Moreover, the termination may be premature, as end loops are often loosely constrained and tend to diverge more than core domain positions. These diverged subsequences may be omitted from the alignment if they decrease the overall similarity score. Therefore the sequence termination signal may be misleading if used simplemindedly. To reduce the sensitivity to sparse signals due to the aforementioned problems with sequence termination, we consider all participating sequences in a position with their evalues (that indirectly indicate alignment's reliability). For every position we calculate right and left termination scores, based on sequences that terminate and originate from that position respectively, by taking the sum of the log of the corresponding evalues. For example if an alignment position has n sequences, of which c terminate at that position and the e-values of the corresponding alignments are $e_1, e_2, ..., e_c$ then the left termination score is defined as

$$E_{left\ termination} = \log(e_1 \cdot e_2 \cdot \cdots \cdot e_c)$$

The left and right termination scores are first smoothed over a window and then multiplied for each position to get the sequence termination score.

2.2.3 Measures of structural flexibility

Indel entropy: In multiple alignment of related sequences positions with indels with respect to the seed sequence indicate regions where there is a certain level of structural flexibility. The larger the number of insertions and the more prominent the variability in the indel length at a position the more flexible we would expect the structure to be in that region. Such structural variability is more likely to occur near a domain boundary where the structure is usually exposed and less constrained. To quantify the structural variability or vulnerability of a position we define the indel entropy based on the distribution of indel lengths as

$$E_g(\mathbf{P}) = -\sum_i p_i \log_2 p_i$$

where the p_i are the various indel lengths seen at a position.

Correlated mutations: Another source of information about the structural flexibility of a position can be obtained from the profile of predicted contacts in a protein. For each sequence position we count the number of pairwise contacts between residues that reside on opposite sides of that position (see also [17]). Minimas in the profile correspond to regions where fewer interactions occur across these sequence positions, implying relatively higher structural flexibility and suggesting a domain boundary.

Contacts between residues in a protein are usually predicted based on correlated mutations. The correlated mutation score between two columns is defined as in [31]. Specifically, the correlation coefficient for two positions k and l is defined as

$$Corr_m(k, l) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{(s(a_{ik}, a_{jk}) - < s_k >)(s(a_{il}, a_{jl}) - < s_l >)}{\sigma_k \cdot \sigma_l}$$

where a_{ik} is the amino acid in position k of sequence i and $s(a, b)$ is the similarity score of amino acids a and b according to the scoring matrix. The term $< s_k >$ is the average similarity in position k and σ_k is the standard deviation. n is the number of sequences that participate in both columns.

To predict a contact based on a correlated mutation score one needs a reliable statistical significance measure to discern true correlations from random coincidental regularities. To assess the statistical significance of correlated mutations we calculated the correlation score for a large collection of random alignment columns. Based on the distribution of the

random scores we associate a $z-score$ with each correlated mutation score[1].

We used the correlated mutation information to design two types of scores. In the first case we considered correlated mutation values that were larger than those in the random distribution as indicating contacts. The number of contacts across every position is then normalized by the total number of possible contacts to generate a contact profile. The other score was based on considering all the values as contacts but weighting them by the z-score to get a weighted profile.

2.2.4 Residue type based measures

Physio-Chemical properties of proteins may also help in predicting domain boundaries since they tend to have different characteristics around domain transition points than in domain core positions. For example, hydrophobic residues tend to cluster inside domain cores with hydrophillic residues occupying more exposed locations in a protein structure and therefore more likely to be in inter-domain regions. Similarly, certain amino acids such as cystines and prolines are crucial in defining protein structure and therefore tend to occur in different frequencies in core domain and inter-domain regions of a protein. In order to exploit these sources of information we defined several measures; for hydrophobicity, molecular weight and for the amino acids cystine, valine, proline and glycine, all believed to be instrumental in defining protein structure. In addition we also used the Rasmol classification of amino-acids to create a set of non-redundant classes that we use as measures (acyclic [ARNDCEQGILKMSTV], aliphatic [AGILV], aromatic [HFWY], buried [ACILMFWV], hydrophobic [AGILMFPWYV], large [REQHILKMFWY], negative [DE], positive [RHK], small [AGS]). For each measure, the score of an alignment column is defined as the average of all residue scores, where residue scores are defined in the range of 0-1 (hydrophobicity and molecular weight are adopted from [32] and class scores are simply defined by the relative frequency of the residues in the class).

2.2.5 Predicted secondary structure information

Protein structure is often studied at the level of secondary structure. Most inter-domain regions are composed of loops while beta strands tend to form sheets that constitute the core of protein domains. Alpha helices and beta sheets in proteins are relatively rigid units and therefore domain boundaries rarely split these secondary structure elements. Indeed, in the study by [33] a domain delineation algorithm was developed that was based on the clustering of secondary structure units. This algorithm was applied to proteins of known structure, and used the available structural information to define the secondary structure elements. However, useful information regarding the secondary structure of a protein can be obtained even when the structure is unknown. We used the neural network based program PSIPRED [34] to predict the secondary structure of the seed protein. The neural network confidence values in the range 0-1 were then used as alpha helix (alpha), beta strand (beta) and coiled region (coil) measures.

[1]Random columns are generated by choosing a root residue at random and mutating it according to transition probabilities, derived from the BLOSUM50 matrix, to generate the other residues in the column.

2.2.6 Intron-exon data

It is well known that the alternative splicing mechanism is used extensively in higher organisms to generate multiple mRNA and protein products from the same DNA strand. This mechanism raises an interesting combinatorial problem. By sampling (and sometimes shuffling) the set of exons encoded in a DNA sequence, the cell generates different proteins that share different numbers of exons.

Intron-exon data at the DNA level is believed to be correlated with domain boundaries [35, 36]. As building blocks, domains are believed to have evolved independently. Therefore it is likely that each domain has a well defined set of exons associated with it. If the product protein is a multi-domain protein we expect exon boundaries to coincide with domain boundaries.

The Intron-exon data was derived from the EID database [37]. Only genes that were experimentally determined (based on the header information) were included in our analysis (a total of 25,130 sequences, and 21,042 entries after eliminating redundancy). Each seed sequence was compared with all the EID sequences, and all significant ungapped matches were recorded. To quantify the likelihood of an exon boundary we use a similar equation as in sequence termination. Specifically, if an alignment position has n sequences, of which c coincide with exon boundaries and the e-values of the corresponding alignments are $e_1, e_2, ..., e_c$ then the exon termination score is defined as

$$E_{exon} = \log(e_1 \cdot e_2 \cdot \cdots \cdot e_c)$$

2.3 Score refinement and normalization

Two additional steps are executed before the measures are fed into the neural network. First, they are smoothed to eliminate random local fluctuations and improve the discrimination power of the measure. The scores are smoothed by calculating the average over a window of size w (the **smoothing factor**). This parameter is optimized to maximize the separation between the two types of positions, as described in the next section.

Second they are normalized to a single scale. To scale all measures to the same units we transformed every score to a z-score based on the distribution of scores along all alignment positions. The normalization is invoked separately for each alignment. The z-score does not only serve as a universal scale but also provides a measure of statistical significance for each position in the alignment, helping to locate extreme a-typical positions.

2.4 Maximizing the information content of scores

To improve domain recognition, the distributions of domain positions and boundary positions (according to each of the domain-information-content measures suggested above) must be well separated. However, it is hardly ever the case that the two distributions are completely disjoint, and the parameters introduced before (the boundary window size x and the smoothing factor w) may greatly affect the separation of these distributions.

To define the best set of parameters we measured the statistical similarity of the two probability distributions for different sets of parameters, and selected the one that maximized separation. To measure statistical similarity we used the Jensen-Shannon (JS) divergence between probability distributions [38]. This is a variation over the KL divergence

measure [39], that is both symmetric and bounded (unlike the KL divergence). Formally, given two (empirical) probability distributions $\mathbf{p}$ and $\mathbf{q}$, for every $0 \leq \lambda \leq 1$, the λ-JS **divergence** is defined as

$$D_{\lambda}^{JS}[\mathbf{p}||\mathbf{q}] = \lambda D^{KL}[\mathbf{p}||\mathbf{r}] + (1-\lambda)D^{KL}[\mathbf{q}||\mathbf{r}]$$

where $\mathbf{r} = \lambda\mathbf{p} + (1-\lambda)\mathbf{q}$ can be considered as the most likely common source distribution of both distributions $\mathbf{p}$ and $\mathbf{q}$, with λ as a prior weight. In our case, the priors for in-domain positions $\mathbf{p}$ and boundary positions $\mathbf{q}$ differ markedly, and λ is set to the prior probability of in-domain positions. We call the corresponding measure the **divergence score**.

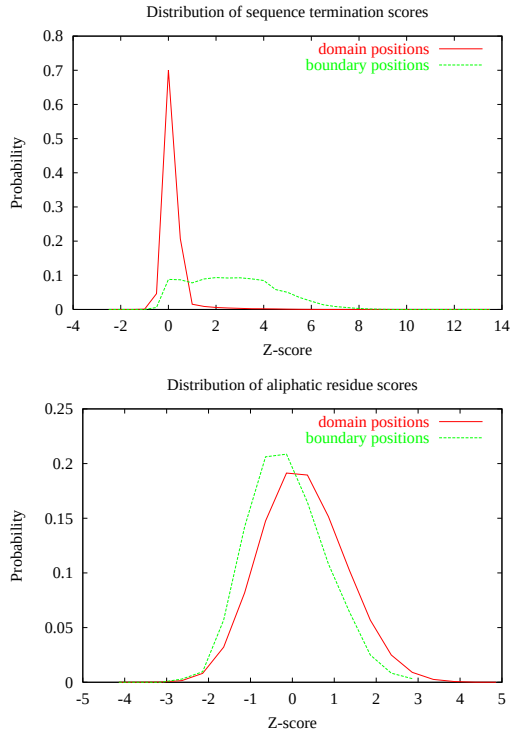

Figure 2: Distributions of sequence termination scores (a) and aliphatic residue scores (b)

Two examples of score distributions are given in Fig. 2. Even measures with identical distributions may be informative in a mutli-variate model, where higher level correlations can generate an effective boundary surface. However, to simplify the final model, only measures that induce different distributions of scores for domain positions and boundary positions are considered for further analysis. The optimal complex decision boundary is learned by training a neural network as described next. The measures that are used to train the network and their Jensen-Shannon divergence are given in Table 1. Although better separation was obtained with individual boundary windows, the final boundary window was uniformly set to $x = 10$ (experiments with smaller window sizes decreased final prediction accuracy) and the smoothing window w was set individually for each score based on the optimization of the Jensen-Shannon divergence.

Score	Smoothing window	Jensen-Shannon divergence
Alpha	4	0.008
Acyclic	8	0.008
Indel Entropy	7	0.010
Consistency	10	0.010
Small	8	0.010
Glycine	10	0.015
Introns	10	0.020
Class Entropy	10	0.024
Weighted Mutation Profile	8	0.034
Proline	10	0.048
Symmetric Correlation	10	0.095
Sequence Termination	7	0.542

Table 1: Jensen-Shannon divergence for different scores. The JS divergence for identical distributions is 0.

2.5 The learning model

Each one of the measures we described in section 2.2 captures some aspects or properties of domain transition signals. To find the optimal combination we trained a neural network over the domain information content scores. A neural network is capable of learning complex non-linear decision boundaries between categories, and therefore seems to be most suited for this task (an alternative model to try would be SVMs). The inputs used were the individual scores in a position, and the output learnt is a number between 0 and 1, where 0 corresponds to a transition point and 1 to a domain. The network was trained using the Matlab neural nets toolbox, on a set of 484 proteins with a validation set of 237 proteins and a test set of 460 proteins. The neural network is a feed-forward network trained using the back propagation algorithm with tangent sigmoid activation function. The best network takes in 12 inputs and has two hidden layers with 10 and 15 neurons respectively. It accurately predicts 94% of the core-domain positions and 88% of the transition points in the test set.

Since a domain transition point is not singular we also tried to learn more complex networks that map multiple inputs (several positions along the sequence) to multiple outputs. However, performance-wise, the basic network (mapping a single sequence position to a single output) performed the best. This might change as more data becomes available.

3. HYPOTHESIS EVALUATION

The neural network does not take into account the information from neighboring positions while making a decision (attempts to learn local neighborhoods in the input space to local neighborhoods in the output space failed to improve the performance). Thus, despite the high rate of accurate predictions for single positions, the final predictions may overly fragment proteins into domains. We experiment with two post-processing setups.

3.1 The simple model

In the simple model, to refine the putative predictions of the neural-net the following two steps are employed. First, a position is predicted as transition point only if a significant fraction of the positions in a window centered around it are predicted as putative transition points by the neural-net (the threshold can be altered to give different levels of accuracy and sensitivity). Second, transition points are listed in decreasing order of reliability (as measured by the depth of the corresponding minima in the smoothed curve) and all

minima that are within a window of 30 amino acids from predicted transitions are rejected.

3.2 The domain-generator model

Given multiple hypotheses, i.e. putative partitions of the query sequence into domains we would like to find the most likely one. The domain-generator model assumes a random generator that moves repeatedly between a domain state and a linker state and emits one domain or transition at a time according to different source probability distributions. Thus the probability of a sequence of domains is given by the product of domain-emission probabilities and the transition probabilities

Formally, we are given a protein sequence S (a multiple alignment) of length L and a possible partition $\mathbf{D}$ of S into n domains $\mathbf{D} = D_1, D_2, .., D_n$ of lengths $l_1, l_2, ..., l_n$ (as suggested by the output of the neural-net). Our goal is to find the most likely model, i.e. the partition that maximizes the posterior probability of the model given the data $P(\mathbf{D}/S)$

We calculate the posterior probability by relying on estimating the likelihood of the data given the partition $P(S/\mathbf{D})$ from the precalculated measures described in section 2.2 and then applying Bayes formula:

$$P(\mathbf{D}/S) = \frac{P(S/\mathbf{D})P(\mathbf{D})}{P(S)}$$

The denominator is fixed for all hypotheses, thus we are looking for the partition that will maximize the product of the likelihood $P(S/\mathbf{D})$ and the prior $P(\mathbf{D})$

To calculate the prior $P(\mathbf{D})$ we have to estimate the probability that an arbitrary protein sequence of length L will consist of d domains of the specific lengths $l_1, l_2, ..., l_n$. What we need to calculate then is

$$P(\mathbf{D}) = P((D_1, l_1)(D_2, l_2)...(D_n, l_n) \ \mathbf{s.t.}\ l_1 + l_2 + .. + l_n = L)$$

This can be estimated from the data by considering known domain partitions of proteins of length L. However, the amount of data available is not enough to accurately estimate these probabilities for all possible partitions. We approximate this probability by using a simplified model; given the length of the protein, the generator selects the number of domains first and then selects the length of one domain at a time, considering the domains that were already generated. For a partition into n domains there are $n!$ possible orderings of the domains, therefore the prior probability of the partition is approximated by

$$P(\mathbf{D}) \simeq Prob(n/L) \cdot \sum_{\pi(l_1,l_2,..,l_n)} P_0(l_1/L)P_0(l_2/L - l_1)...P_0(l_{n-1}/L - \sum_{1}^{n-2} l_i)$$

where $Prob(n/L)$ is the prior probability that a sequence of length L constitutes of n domains and $P_0(l_i/L)$ is the prior probability to emit a domain of length l_i given a sequence of length L. The term $\pi(l_1, l_2, .., l_n)$ denotes all possible permutations of $l_1, l_2, .., l_n$. The prior probabilities $P_0(l_i/L)$ are approximated by $P_0(l_i)$, normalized to the relevant range $[0..L]$, and are estimated from the distribution of domain lengths in the SCOP database[2].

[2]Ideally, we would like to use $P_0(l_i/L)$. However, the SCOP data set is very noisy and the resulting distributions are heavily biased towards the domain definitions in SCOP. We are currently working on improving these estimates.

The second term, $Prob(n/L)$ is given by $Prob(n/L) = Prob(n, L)/P(L)$ where $Prob(n, L)$ is estimated by the $(n-1)$th order sum

$$Prob(n, L) = \sum_{1}^{L} P_0(x_1) \sum_{1}^{L} P_0(x_2)... \sum_{1}^{L} P_0(x_{n-1}) \cdot \cdot P_0(L - x_1 - x_2 - ... - x_{n-1})$$

and $P(L)$ is simply given by the complete probability formula

$$P(L) = \sum_{i=i}^{L} Prob(i, L)$$

To calculate the likelihood of the data given the model $P(S/\mathbf{D})$ we use the probabilities of the observed scores given the domain structure as predicted by the neural-net. We consider the individual domains and the transitions between domains (the linkers) as two different sources. Each source induces a unique probability distribution over the domain-information content scores (see section 2.2). Specifically, given the model $\mathbf{D}$ that partitions the sequence S into n domains and $n-1$ transitions $D_1, T_1, D_2, T_2, ..T_{n-1}, D_n$ that correspond to the subsequences $s_1, t_1, s_2, t_2, ...t_{n-1}, s_n$ we estimate the likelihood by

$$\begin{aligned} P(S/\mathbf{D}) &= P(S/D_1, T_1, D_2, ..T_{n-1}, D_n) \\ &= P(s_1/D_1)P(t_1/T_1)P(s_2/D_2) \cdot \\ &\quad \cdot P(t_2/T_2)...P(t_{n-1}/T_{n-1})P(s_n/D_n) \end{aligned}$$

where we already employed the assumption that the domains are independent of each other. Each one of the terms $P(s_i/D_i)$ and $P(t_j/T_j)$ is a product over the probabilities of the individual positions. The probability on an individual position j in domain i is estimated by the joint probability distribution of the 12 features that are used in our system

$$P(s_{ij}/D_i) = P(f_1, f_2, ...f_{12}/D_i)$$

However, estimating this probability is impractical given the amount of data we have. On the other hand, given the correlation between scores (see section 4.1) the independence assumption for the individual scores does not hold. Therefore we adopt an intermediate approach. We start by writing the exact formulation of the joint probability distribution of k random variables $X_1, X_2, .., X_k$ using the expansion

$$\begin{aligned} P(X_1, X_2, .., X_k) &= P(X_1)P(X_2/X_1)P(X_3/X_1, X_2) \cdots \\ &\quad \cdots P(X_k/X_1, X_2, .., X_{k-1}) \end{aligned}$$

where the random variables can be ordered in an arbitrary order. To derive a sensible approximation of these probabilities we use first-order dependencies[3] and employ the following heuristic. For each pair of random variables X, Y we calculate the distance between the joint probability distribution and the product of the marginal probability distributions

$$DEPEN(X, Y) \equiv Dist(P_{XY}, P_X P_Y)$$

This distance (measured either using the l_1 norm or the JS divergence measure) is a measure of the dependency between the two variables. The larger it is, the more dependent are

[3]Pair statistics can be calculated quite reliably from our data set, but the data is too sparse to derive reliable estimates of higher order statistics

the variables (one might also consider using the mutual information measure instead). We sort all pairs based on their distance and pick the most dependent one first (denoted by Y_1 and Y_2) to start the expansion

$$P(X_1, X_2, .., X_k) = P(Y_1)P(Y_2/Y_1).....$$

The next terms are selected based on their strongest dependency with variables that are already used in the expansion. Thus

$$Y_3 = \arg\max_Y \{\max\{DEPEN(Y, Y_1), DEPEN(Y, Y_2)\}\}$$

Denote by $Z = PILLAR(Y)$ the random variable that Y is most dependent on (of the random variables that are already in the expansion), then of all possible dependencies involving Y_3 we pick $P(Y_3/PILLAR(Y_3))$ and add it to the expansion

$$P(X_1, X_2, .., X_k) = P(Y_1)P(Y_2/Y_1)P(Y_3/PILLAR(Y_3)).....$$

The procedure continues until all variables are accounted for. This heuristic attempts to minimize the errors that are introduced by relaxing the dependency assumption to a first order dependency by maximizing the support for each random variable we introduce in the expansion. Thus, highly correlated variables affect the total probability only marginally, while under the independence assumption they might introduce a substantial error (other, alternative methods for approximating the joint probability distribution from the marginal distributions are described in [41] and [42] and we are currently testing their effect on our estimates). Note that the expansion for domain regions can be different from the expansion for linker regions, as the source distributions differ. However, once the two expansions (for domains and linkers) are defined based on the pair statistics, the same expansions are used for all domains and all linkers.

Finally, given a set of N putative transition points (as described in section 3), our algorithm enumerates all possible combinations of transition points to form 2^N possible partitions (hypotheses). For each partition we calculate the posterior probability and eventually output the most likely one. The whole calculation is very fast. For example, for a protein of length $L = 300$ and set of $N = 10$ possible transition points, the algorithm will output the most probable hypothesis in a matter of minutes.

4. RESULTS

To test our approach we run our system on a subset of 460 proteins that were excluded from the training set. For each of these proteins the prediction was compared to that of SMART [15], Pfam [12, 13], ProDom [9] and Tigr [14], based on the information provided by InterPro [43] as well as predictions from DOMO [11] obtained by running BLAST searches against the DOMO database.

Since the predictions obtained from other systems are often incomplete for the seed proteins in our test set, we needed to design an evaluation procedure that would have different scores for accuracy and coverage. In addition, the predictions may disagree with SCOP on the number of domains in the seed protein. Therefore one needs to define an algorithm for associating predicted transition points with their most probable SCOP counterparts and vice versa. The simplest choice is to assign every transition point that is being considered to the closest reference transition point. Here we adopt this model[4] and define the following four measures.

Distance accuracy. This measure evaluates predictions by using SCOP transition points as reference. For each seed protein we calculate the average distance of the predicted transitions from their associated SCOP transition points. The final value that is reported is the average distance over all proteins in the test set.

Distance sensitivity. This measure assesses the sensitivity in detecting true domain boundaries by using the predicted transitions as reference. The average distance of SCOP transitions from the associated predicted transitions is calculated for each protein, with the value reported being the average of this distance over all proteins in the test set.

Selectivity. For this measure we consider predictions that are within $x = 10$ residues of a SCOP transition as being correct with the final value reported being the percentage of predictions that are considered correct for the entire set.

Coverage. Analogous to accuracy, SCOP transitions that are associated with a predicted transition point within $x = 10$ residues are considered successfully predicted. The percentage of correctly predicted SCOP transitions for the entire set is reported.

The results of our tests are summarized in Table 2. To assess the impact of the different components of our model two different sets of numbers are given; for the simple model and for the domain-generator model as described in section 3. Note that our method outperformed other methods in terms of the defined measures, even the manually calibrated Pfam, while being fully automatic[5].

	Number of predictions	Accuracy/Sensitivity (in residues)	Selectivity/Coverage (percentages)
simple model	460	26/7	66/82
domain-generator	460	27/6	63/83
HMMPfam	441	29/14	43/65
BlastDomo	252	17/70	22/12
HMMSmart	172	12/73	27/17
BlastProDom	123	8/90	30/6
HMMTigr	51	2/96	33/1

Table 2: Performance evaluation results.

.

It is important to note that although the domain-generator model does not improve over the simple model, as measured by the four performance indices described above, it is a better model overall. Specifically, the domain-generator model provides us with a critical statistical framework for assessing alternative, competing hypotheses. The model can be used to assign a confidence value to each hypothesis, and by comparing these confidence values (between the best hypothesis and the next best hypothesis or the set of all other hypotheses) one can define a significance measure and associate it with the output hypothesis. In cases where the differences between competing hypotheses are insignificant, one might want to consider also alternative hypotheses.

[4]We are currently working on developing a more sophisticated association scheme.

[5]Pfam is actually using the SCOP definitions (when available) to determine their domain definitions. Their performance therefore may not be as good over an independent data set.

4.1 Examples

The overall performance of our method shows that the model is capable of learning even subtle signals that indicate domain boundaries. Our first example is a four domain protein that was predicted accurately for all its domains. This is the PDB protein 2gep, 497 residues long. The protein is partitioned by SCOP into four domains that correspond to positions 8-72, 73-272, 273-352 and 353-497. Our prediction suggests transition points at positions 75, 270 and 352 (see Fig. 3) within 3 residues from SCOP definitions. These positions are correlated with strong sequence termination and correlated mutations signals. It should be noted that the sequence termination measure is quite noisy in this case because of a complex alignment, and there are incorrect signals at positions 160, 240, 300, 420, 440 and 460. However, our system successfully rejected these transitions. For comparison, PFam predicts three domains between positions 1-67, 273-345 and 356-425 (nitrite/sulfide reductase ferrodoxin-like half domain at 1-67 and 273-345, and nitrite and sulfite reductase at 356-425). DOMO predicts a single domain (positions 1-481). No predictions are available from ProDom, SMART or Tigr.

Figure 3: (a) Domain definitions for 2gep. Our method predicts four domains. The transition points are marked by their residue number (note that positions are offset by 81 since the first residue in the PDB file is numbered as residue 81).

Another example where our method correctly predicted all the domain transition points is for the protein 1a8y. However, in this case none of the other sequence-based predictions (including Pfam) were able to partition the protein correctly. This protein is 367 amino acids long, and according to SCOP it consists of three independent domains, between positions 3-126, 127-228 and 229-347 (see Fig. 4). No clear distinction is made in SCOP regarding their different functional role. Our prediction locates domain boundaries at positions 125 and 225, within 4 residues from the SCOP definition.

According to Pfam, the main domain (Calsequestrin) is located between positions 1 and 362, transcending the structural domain boundaries. Domo predicts one domain between positions 1 and 335. No predictions are available from ProDom, SMART or Tigr. Detailed analysis of our system in this case reveals sequence termination signals at positions 30, 120, 130, 210, 230, 270 and 280 (with signals at 230, 280 and 120 being dominant), three major entropy peaks at 70, 120 and 210 and a proline rich region between positions 210 and 240 (that corresponds to a sharp turn in the structure). Major correlation troughs are also detected at positions 40, 120-140 and 260 (data is not presented graphically because of limited space).

Figure 4: (a) Domain definitions for 1a8y. In this case a mosaic of signals (sequence termination, entropy, correlation and residue type) is integrated by our system into three precise predictions that are in perfect agreement with structural definitions.

4.2 Suggested novel partitions

The list of proteins on which our method failed to correctly predict domain boundaries as defined by SCOP revealed interesting cases. Many of them raise serious questions about the validity of the SCOP definitions. For example, PDB protein 1acc (735 amino acids long) is defined as a single domain in SCOP. Our analysis suggests three domains, at positions 1-167, 168-586 and 587-735 (see Fig. 5). As the figure illustrates, this partition seems to better satisfy the definition of a domain as a compact, independent foldable unit. Moreover, given the distribution of domain sizes in proteins (see section 3.2), it is not very likely to have protein domains that are longer than 700 amino acids, further supporting our hypothesis. For comparison, Pfam detects one domain at positions 103-544 (PF03495 Clostridial Binary exotoxin B) and Domo predicts two domains at positions 1-647 and 648-735. No predictions are available from ProDom, SMART or Tigr.

Figure 5: (a) Domain definitions for 1acc. SCOP defines this protein as a single domain. Our analysis suggests three compact units.

In this case we get clean and strong sequence termination signals at positions 150, 170, 590 and 610 and a remark-

ably consistent alignment between positions 170 and 580. This signal is reinforced by other measures: the hydrophobic curve has three major troughs at 170, 290 and 570, insertion entropy has major peaks at 180, 310 and 560 and correlation is pretty low around 200, 280 and 590.

5. DISCUSSION

In this paper we presented a novel method for detecting the domain structure of a protein from sequence information alone, by utilizing the information in sequence databases. The query sequence is compared with all the sequences in the database and the resulting alignment is processed fully automatically in search for domain transition signals. There are several novel elements in out method. First, our method uses multiple scores. Some of the scores we designed are variations on measures that were suggested in earlier studies (e.g sequence participation and correlation scores were used in DOMO, ProDom and PASS and correlated mutations were used in Rigden's work). However, we introduce many novel scores based on analysis of basic sequence properties or predicted properties, scores that are calculated from multiple alignments, and scores that are extracted from external resources such as intron-exon data. Second, information theory principles are used to optimize the scores and select the subset that maximizes the domain information content. Third, a neural network is trained to learn a non-linear mapping from the original scores to a single output[6]. Finally, a probabilistic domain-generator model is developed to assess multiple hypotheses and predict the most likely one. This multi-stage system is not only robust to alignment inaccuracies, but it can also tolerate partial information. It can be extended and generalized to include other types of scores. Most importantly, our method suggests for the first time a rigorous model that can test all possible hypotheses and output the one that is most consistent with the data. We also developed an evaluation framework that hopefully will provide a clearer understanding of the strengths and weaknesses of the algorithms that have been designed so far and thus aid in the design of better algorithms. Moreover, our domain-generator model can associate a statistical significance score for every hypothesis, thus enabling us to compare different hypotheses by the same method or even different hypotheses by several different methods.

We trained and tested our method on what is considered to be the gold standard in protein structure classification, the SCOP database of protein domains. Our method was better than the best manual methods currently available while being fully automatic. One should keep in mind that SCOP is a man-made classification, and the definitions of domains do not necessarily conform with "nature's definitions". Indeed many of our supposedly errors seem to make sense when inspected visually.

We are already considering several variations to the model described here. Although our algorithm is not overly sensitive to alignment accuracy, obviously better multiple alignment algorithms are expected to improve the performance. Another possible improvement is the integration of a weighting scheme into the multiple alignment. Currently all sequences are weighted equally. However, due to the biased representation of protein families in sequence databases and the nature of sequence comparison algorithms, diverged sequences that might provide us with crucial information about domain boundaries are usually underrepresented in these alignments. To eliminate this bias one should decrease the weight of highly similar sequences and increase the weight of highly diverged sequences. This modification in currently underway. Hopefully these variations will improve performance even more.

[6]Early attempts to use a linear system failed to provide satisfactory performance.

6. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant No. 0133311 to Golan Yona.

7. REFERENCES

[1] Rose, G. D. (1979). Hierarchic organization of domains in globular proteins. *J. Mol. Biol.* **134**, 447-470.

[2] Lesk, A. M. & Rose, G. D. (1981). Folding units in globular proteins. *Proc. Natl. Acad. Sci. USA* **78**, 4304-4308.

[3] Holm, L. & Sander, C. (1994). Parser for protein folding units. *Proteins* **19**, 256-268.

[4] Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **247**, 536-540.

[5] Yona, G. & Levitt, M. (2000). Towards a complete map of the protein space based on a unified sequence and structure analysis of all known proteins. *In the proceedings of ISMB 2000*, 395-406, AAAI press, Menlo Park.

[6] Kuroda, Y., Tani, K., Matsuo, Y. & Yokoyama, S. (2000). Automated search of natively folded protein fragments for high-throughput structure determination in structural genomics. *Protein Sci.* **9**, 2313-2321.

[7] George, R. A. & Heringa, J. (2002). Protein domain identification and improved sequence similarity searching using PSI-BLAST. *Proteins* **48**, 672-681.

[8] Gouzy, J., Corpet, F. & Kahn, D. (1999). Whole genome protein domain analysis using a new method for domain clustering. *Comput Chem.* **23**, 333-340.

[9] Sonnhammer, E. L. L. & Kahn, D. (1994). Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci.* **3**, 482-492.

[10] Park, J. & Teicmann, S. A. (1998). DIVCLUS: an automatic method in the GEANFAMMER package that finds homologous domains in single- and multi-domain proteins. *Bioinformatics* **14:2**, 144-150.

[11] Gracy, J. & Argos, P. (1998). Automated protein sequence database classification. I. Integration of copositional similarity search, local similarity search and multiple sequence alignment. II. Delineation of domain boundries from sequence similarity. *Bioinformatics* **14:2**, 164-187.

[12] Sonnhammer, E. L., Eddy, S. R., Durbin, R. (1997). Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins* **28**, 405-420.

[13] Bateman, A., Birney, E., Durbin, R., Eddy, S. R., Finn R. D., & Sonnhammer E. L. (1999). Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucl. Acids Res.* **27**, 260-262.

[14] Haft, D. H., Loftus, B. J., Richardson, D. L., Yang, F., Eisen, J. A., Paulsen, I. T. & White, O. (2001). TIGRFAMs: a protein family resource for the functional identification of proteins. *Nucl. Acids Res.* **29**, 41-43.

[15] Ponting, C. P., Schultz, J., Milpetz, F. & Bork, P. (1999). SMART: identification and annotation of domains from signalling and extracellular protein sequences. *Nucl. Acids Res.* **27**, 229-232.

[16] George, R. A. & Heringa, J. (2002). SnapDRAGON: a method to delineate protein structural domains from sequence data. *J. Mol. Biol.* **316**, 839-851.

[17] Rigden, D. J. (2002). Use of covariance analysis for the prediction of structural domain boundaries from multiple protein sequence alignments. *Protein Eng.* **15**, 65-77.

[18] Guan, X. & Du, L. (1998). Domain identification by clustering sequence alignments. *Bioinformatics* **14**, 783-788.

[19] Wheelan, S. J., Marchler-Bauer, A. & Bryant, S. H. (2000). Domain size distributions can predict domain boundaries. *Bioinformatics* **16**, 613-618.

[20] George, D. G., Barker, W. C., Mewes, H. W., Pfeiffer, F. & Tsugita, A. (1996). The PIR-International protein sequence database. *Nucl. Acids. Res.* **24**, 17-20.

[21] Bairoch, A. & Apweiler, R. (1999). The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucl. Acids Res.* **27** 49-54.

[22] Hubbard, T. J., Ailey, B., Brenner, S. E., Murzin, A. G. & Chothia, C. (1999). SCOP: a Structural Classification of Proteins database. *Nucl. Acids Res.* **27**, 254-256.

[23] Westbrook, J., Feng, Z., Jain, S. et al. (2002). The Protein Data Bank: unifying the archive. *Nucl. Acids. Res.* **30**, 245-248

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

[25] Yona, G., Linial, N. & Linial, M. (1999). ProtoMap: Automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins*, **37**, 360-378.

[26] Henikoff, J. G. & Henikoff, S. (1996). Using substitution probabilities to improve position-specific scoring matrices. *Comp. App. Biosci.* **12:2**, 135-143.

[27] Hobohm, U. & Sander, C. (1995). A sequence property approach to searching protein database. *J. Mol. Biol.* **251**, 390-399.

[28] Ferran, E. A., Pflugfelder, B. & Ferrara P. (1994). Self-Organized Neural Maps of Human Protein Sequences. *Protein Sci.* **3**, 507-521.

[29] Csiszr, I. Information Theoretic Methods in Probability and Statistics. From citeseer.nj.nec.com

[30] Henikoff, S. & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA* **89**, 10915-10919.

[31] Pazos, F., Helmer-Citterich, M., Ausiello, G. & Valencia, A. (1997). Correlated mutations contain information about protein-protein interaction. *J. Mol. Biol.* **271**, 511-523.

[32] Black, S.D. & Mould, D.R. (1991). Development of Hydrophobicity Parameters to Analyze Proteins Which Bear Post or Cotranslational Modifications. *Anal. Biochem.* **193**, 72-82.

[33] Sowdhamini, R. & Blundell, T. L. (1995). An automatic method involving cluster analysis of secondary structures for the identification of domains in proteins. *Protein Sci.* **4**, 506-520.

[34] McGuffin, L. J. , Bryson, K. & Jones, D. T. (2000). The PSIPRED protein structure prediction server. *Bioinformatics* **16**, 404-405.

[35] Gilbert, W. & Glynias, M. (1993). On the ancient nature of introns. *Gene* **135**, 137-144.

[36] Gilbert, W., de Souza, S. J. & Long, M. (1997). Origin of genes. *Proc. Natl Acad. Sci. USA* **94**, 7698-7703.

[37] Saxonov, S. , Daizadeh, I. , Fedorov, A. & Gilbert, W. (2000). EID: the Exon-Intron Database-an exhaustive database of protein-coding intron-containing genes. *Nucl. Acids Res.* **28**, 185-190.

[38] Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Trans. Info. Theory* **37:1**, 145-151.

[39] Kullback, S. (1959). "Information theory and statistics". John Wiley and Sons, New York.

[40] El-Yaniv, R., Fine, S. & Tishby, N. (1997). Agnostic classification of markovian sequences. *Advances in Neural Information Processing Systems* **10**, 465-471.

[41] Ireland, C. T. & Kullback, S. (1968). Contingency tables with given marginals. *Biometrika* **55**, 179-189.

[42] Pearl, J. (1997). "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference." Morgan Kaufmann Publishers Inc., San Mateo, California.

[43] Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M., Bucher, P., Cerutti, L., Corpet, F., Croning, M. D., Durbin, R., Falquet, L., Fleischmann, W., Gouzy, J., Hermjakob, H., Hulo, N., Jonassen, I., Kahn, D., Kanapin, A., Karavidopoulou, Y., Lopez, R., Marx, B., Mulder, N. J., Oinn, T. M., Pagni, M., Servant, F., Sigrist, C. J. & Zdobnov, E. M. (2001). The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucl. Acids Res.* **29**, 37-40.