DOMpro: Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks

Jianlin Cheng Michael J. Sweredoski Pierre Baldi

Institute for Genomics and Bioinformatics
School of Information and Computer Science
University of California Irvine
Irvine, CA 92697, USA
{*jianlinc,msweredo,pfbaldi*}*@ics.uci.edu*

Protein domains are the basic macroscopic units of protein tertiary structure. Being able to parse protein chains into different domains is important for protein classification and for understanding protein structure, function, and evolution. Here we use machine learning algorithms, in the form of recursive neural networks, to develop a protein domain predictor called DOMpro. DOMpro predicts protein domains using a combination of evolutionary information in the form of profiles, predicted secondary structure, and predicted relative solvent accessibility. DOMpro is trained and tested on a curated dataset derived from the CATH database. DOMpro correctly predicts the number of domains for 69% of the combined dataset of single and multi-domain chains. Of the single domain proteins, 79% are correctly predicted as having no domain boundaries. The total number of domains is correctly predicted for 43% of the multi-domain proteins. DOMpro is able to correctly predict both the domain number and domain boundary location for 25% of the two domain chains. DOMpro is a member of the SCRATCH suite of predictors available through *http://www.igb.uci.edu/servers/psss.html.*

Keywords: protein structure prediction, domain, recursive neural networks

1. Introduction

Domains are considered the basic macroscopic units of protein tertiary structure. Most definitions of domains rely on different criteria ranging from the ability to fold independently, to evolutionary conservation, to discrete functionality [7]. A domain can span an entire polypeptide chain or be a subunit of a polypeptide chain that can fold into a stable tertiary structure independently of any other domain [12]. While many domains are comprised of a single continuous polypeptide segment, in some cases domains may be comprised of several discontinuous segments.

The identification of domains is an important step for protein classification and for the study/prediction of protein structure, function, and evolution. The topology of secondary structure elements in a domain is used by human experts or automated systems in structural classification databases such as FSSP-Dali Domain Dictionary [8,9], SCOP [17], and CATH [19]. The prediction of protein tertiary structure, especially *ab initio* prediction, can be improved by using domain boundary information [5] and applying prediction methods separately to each domain. However, the identification of protein domains based on sequence alone remains a challenging problem.

A number of methods have been developed to identify protein domains starting from the primary sequence. Some of these methods use a sequence alignment approach whereby domains are identified by aligning the target sequence against sequences in a domain classification database [14]. Other methods use alignments of secondary structure [15]. In these methods, domains are assigned by aligning the predicted secondary structure of a target sequence against the secondary structure of chains in CATH with known domain boundaries. Tertiary structure folding approaches such as SnapDRAGON [6] average several hundred predictions obtained from coarse *ab initio* simulations of protein folding for a given sequence to assign its domain content. One drawback to such approaches is that they are very computationally intensive. Statistical meth-

ods such as Domain Guess by Size (DGS) [24] predict the likelihood of domain boundaries within a given sequence based on statistical distributions of chain and domain lengths.

The prediction of domains using machine learning techniques is aided by the availability of large, high quality domain classification databases such as CATH, SCOP and DaliDD. Two recently published algorithms attempt to predict domain boundaries using neural networks [13,18]. The networks used by Nagarajan and Yona (2004) incorporate the position specific physiochemical properties of amino acid and predicted secondary structure. Liu and Rost (2004) use neural networks with amino acid composition , positional evolutionary conservation, and predicted secondary structure and solvent accessibility.

Here we describe DOMpro, a novel machine learning approach for predicting domains, which uses profiles along with predicted secondary structure and solvent accessibility in a 1D-recursive neural network (1D-RNN). These networks are also used for prediction of the secondary structure and solvent accessibility [21,22] in our SCRATCH suite of servers [2]. Unlike previous neural network-based approaches [13,18], the direct use of profiles in DOMpro is based on the assumption that sequence motifs and their level of conservation in the boundary regions are different from those found in the rest of the protein. The final assignment of protein domains is the result of post-processing and statistical inference on the output of the recursive neural network.

2. Methods

2.1. *Data*

DOMpro is trained and tested on a curated dataset derived from annotated domains in the CATH domain database, version 2.5.1. Because the CATH database contains only the sequences of domain regions, sequences from the PDB [4] must be incorporated to reconstruct entire chains. Once the chains are reconstructed, short sequences (< 40 residues) are filtered out.

We then use UniqueProt [16] to reduce the sequence redundancy in the dataset. We ensure that no pair of sequences in the dataset have a HSSP value greater than 5. The HSSP value between two sequences is a measure of their similarity and takes into account both sequence identity and sequence length. A HSSP value of 5 corresponds roughly to a sequence identity of 25% in a global alignment of length 250.

Finally, secondary structure classification and relative solvent accessibility are predicted for each chain using SSpro and ACCpro [21,22,2]. Using predicted, rather than true secondary structure and solvent accessibility which are easily-obtainable by the DSSP program [11], introduces additional robustness in the predictor, especially when it is applied to sequences with little or no homology to sequences in the PDB. To leverage evolutionary information, PSI-BLAST [1] is used to generate profiles by aligning all chains against the Non-Redundant (NR) database, as in [10,23,21].

After redundancy reduction, our curated dataset contains 355 multi-domain chains and 963 single domain chains. The ratio of single to multi-domain chains reflects the skewed distribution of single domain chains in the PDB [4]. Figure 1 shows the frequency of single and multiple domain chains in the redundancy-reduced dataset. Figure 2 shows the distribution of chain lengths among single and multi-domain chains.

Because the recursive neural networks are trained to recognize domain boundaries, only multi-domain proteins are used during the training process. During the training and testing of the neural networks on multi-domain proteins, ten fold cross-validation is used. Additional testing is performed on single domain proteins using models trained with multi-domain proteins.

2.2. *Input and Output of Neural Networks*

The problem of predicting domain boundary can be viewed as a binary classification problem for each residue along one dimensional (1D) protein chain. The residue at position i is labelled as

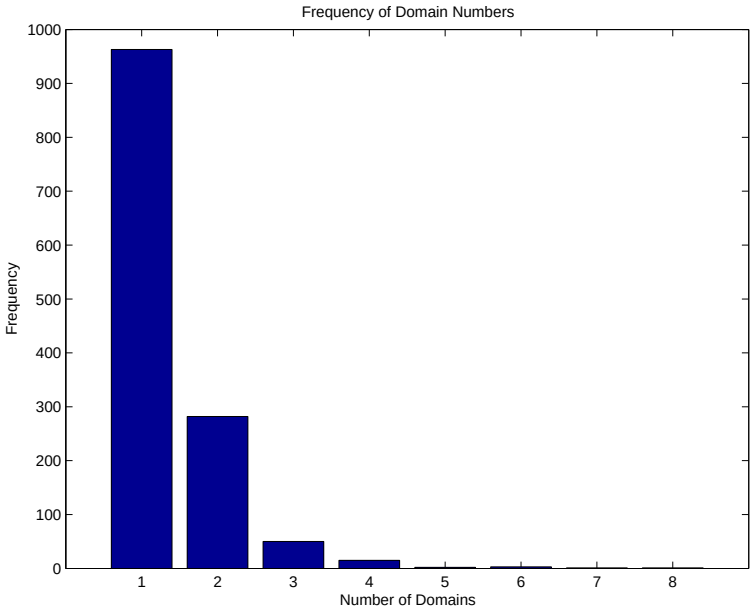


Fig. 1. Frequency of single and multi-domain chains in the redundancy-reduced dataset.

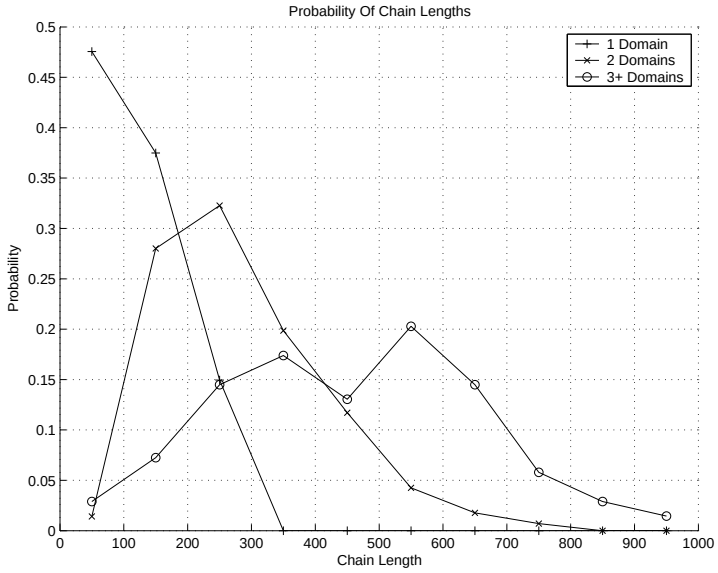


Fig. 2. Distribution of the lengths of single and multi-domain chains in the redundancy-reduced dataset.

within domain boundary regions or not. Specifically, the target class for each residue is defined as follows. Residues within 20 amino acids of a domain boundary are considered domain boundary residues and all other residues are considered non-boundary residues. A variety of machine learning methods can be applied to this problem, such as probabilistic graphical models, kernel methods, and neural networks. DOMpro employs 1D recursive neural networks (1D-RNN)[2]. For each chain, our input is 1D array I, where the size of I is equal to the number of residues in the chain and each entry I_i is a vector of dimension 25 encoding the profile as well as secondary structure and relative solvent accessibility at position i. Specifically, twenty of the values are real numbers which correspond to the amino acid profile. The other five values are binary. Three

of the values correspond to the predicted secondary structure class (Helix, Strand, or Coil) of the residue and the other two correspond to the predicted relative solvent accessibility of the residue (i.e., under or over 25% exposed).

The training target for each chain is a 1D binary array T, whereby each T_i equals 0 or 1 depending on whether residue at position i is within boundary region or not. Neural networks or other machine learning methods can be trained on the data set to learn a mapping from the input array I onto an output array O, whereby O_i is the predicted probability that residue at position i is within domain boundary region. The goal is to make the output O as close as possible to the target T.

2.3. *The Architecture of 1D-Recursive Neural Networks (1D-RNNs)*

The architecture of 1D-RNNs used in this study is derived from the theory of probabilistic graphical models, but use a neural network parameterization to speed up belief propagation and learning [2]. 1D-RNNs combine the flexibility of Bayesian networks with the fast, convenient, parameterization of artificial neural networks without the drawbacks of standard feedforward neural networks with fixed input size. Under this architecture, the output O_i depends on the entire input I instead of a local fixed-width window centered at position i. Thus, 1D-RNNs can handle inputs with variable length and allow classification decisions to be made based on contextual long-ranged information outside of the traditional local input window.

The architecture of the 1D-RNN is described in figures 3 and 4 and is associated with a set of input variables I_i, a forward H_i^F and backward H_i^B chain of hidden variables, and a set O_i of output variables. In terms of probabilistic graphical models (Bayesian networks), this architecture has the connectivity pattern of an input-output HMM [3], augmented with a backward chain of hidden states. The backward chain is of course optional and used here to capture the spatial, rather than temporal, properties of biological sequences.

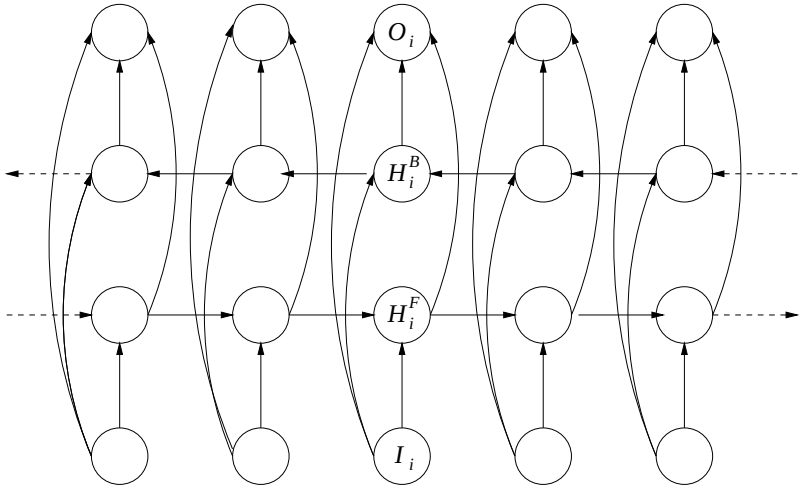


Fig. 3. 1D-RNN associated with input variables, output variables, and both forward and backward chains of hidden variables.

The relationship between the variables can be modeled using three separate neural networks to compute the output, forward, and backward variables respectively. These neural networks are replicated at each position i;(i.e., weight sharing). One fairly general form of weight sharing is to assume stationarity for the output, forward, and backward networks, which finally leads to a 1D-RNN architectures, previously named bidirectional RNN architecture (BRNN), implemented using three neural networks $\mathcal{N}_O$, $\mathcal{N}_F$, and $\mathcal{N}_B$ in the form

$$\begin{aligned} O_i &= \mathcal{N}_O(I_i, H_i^F, H_i^B) \\ H_i^F &= \mathcal{N}_F(I_i, H_{i-1}^F) \\ H_i^B &= \mathcal{N}_B(I_i, H_{i+1}^B) \end{aligned} \tag{1}$$

as depicted in Figure 4. In this form, the output depends on the local input I_i at position i, the forward (upstream) hidden context $H_i^F \in \mathbb{R}^n$ and the backward (downstream) hidden context $H_i^B \in \mathbb{R}^m$, with usually $m = n$. The boundary conditions for H_i^F and H_i^B can be set to 0, i.e. $H_0^F = H_{N+1}^B = 0$ where N is the length of the sequence being processed. Alternatively these boundaries can also be treated as a learnable parameter. Intuitively, we can think of $\mathcal{N}_F$ and $\mathcal{N}_B$ in terms of two "wheels" that can be rolled along the sequence. For the prediction at position i, we roll the wheels in opposite directions starting from the N- and C- terminus and up to position i. Then we combine the wheel outputs at position i together with the input I_i to compute the output prediction O_i using $\mathcal{N}_O$.

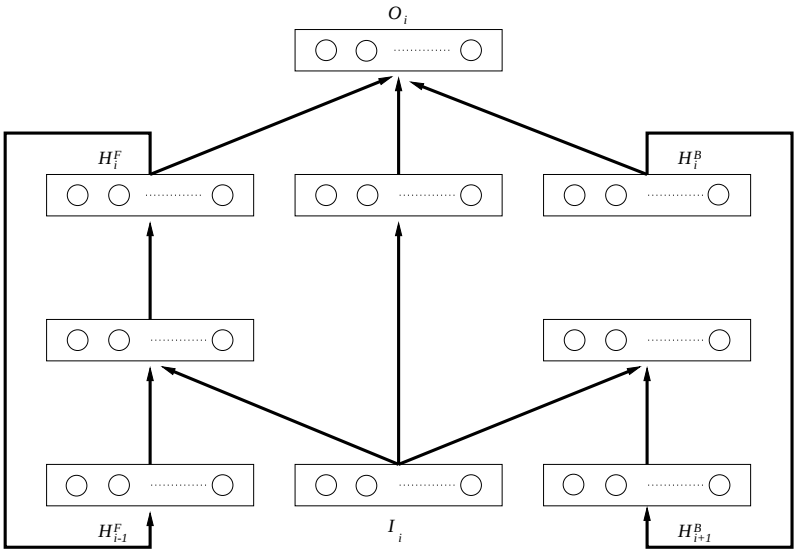


Fig. 4. A 1D-RNN architecture with a left (forward) and right (backward) context associated with two recurrent networks (wheels).

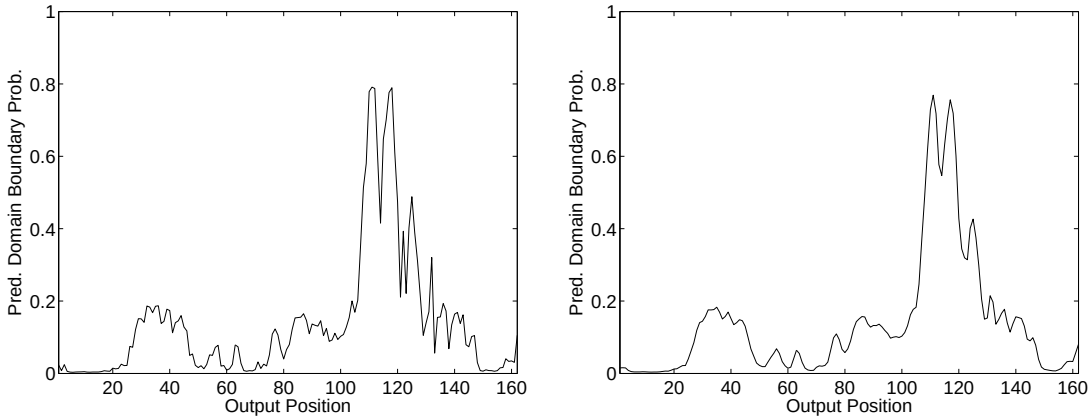
The output O_i for each residue position i is computed by two normalized-exponential units, equivalent to one logistic output unit. The error function is the relative entropy between the true distribution and the predicted distribution.

All the weights of the 1D-RNN architecture, including the weights in the recurrent wheels, are trained in supervised fashion using a generalized form of gradient descent on the error function, derived by unfolding the wheels in space.

2.4. *Post-Processing of 1D-RNN Output*

The raw output from the 1D-RNN is quite noisy (See Figure 5). DOMpro uses smoothing to help correct for the random noise that is the result of false positive hits. The smoothing is accomplished by averaging over a window of length three around each position. Figure 5 shows how this smoothing technique helps to reduce the noise found in the raw output of the 1D-RNN. After smoothing, a domain state (boundary/not boundary) is assigned to each residue by thresholding the networks output at .5.

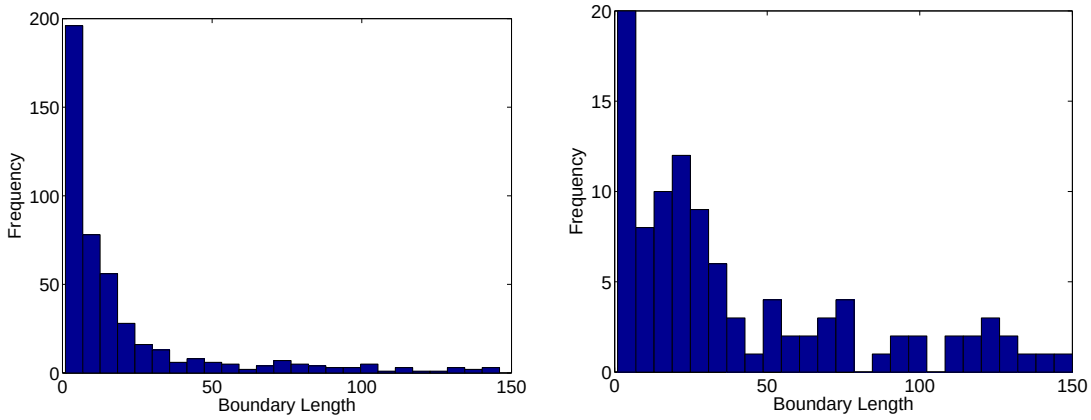
While smoothing the neural network output helps correct for random spikes, it does not necessarily create the long, continuous segments of boundary residues that are required for domain assignment. Therefore, further inference on the output is required.



Raw output from 1D-RNN

Smoothed output from 1D-RNN with window of width 3

Fig. 5. Smoothing of raw output from 1D-RNN



False positive boundaries

True boundaries

Fig. 6. Histogram of length distributions of false and true positive boundary regions

DOMpro infers domain boundary regions from residues predicted as domain boundaries by pattern matching on the discretized output. Any section of the output which matches the regular expression pattern ((B+N{0,m})*B+) is considered a domain boundary region, where B is a predicted boundary residue, N is a predicted non-boundary residue and m is the maximum separation between two boundary residues which should be merged into one region.

Once DOMpro has inferred all possible domain boundary regions, we need to identify false positive domain boundary regions. DOMpro considers the boundary region's length a measure of its signal strength. Figure 6 shows that there is a clear difference between the length distributions of true domain boundary regions and false domain boundary regions. Based on these statistics, domain boundary regions shorter than three residues are considered false positive hits and are ignored. The target sequence is then cut into domain segments at the middle residue of each boundary region. A target sequence with no predicted domain boundaries is classified as a single domain chain.

The final step of DOMpro is to assign domain numbers to each predicted domain segment. One naive method is to assign each domain segment to a separate domain. However, this method fails to identify discontinuous domains. One possible strategy to overcome this problem is to

combine predicted domain segment information with predicted contact map information in order to assign domain numbers. To handle discontinuous domains comprising two ore more disjoint segments, the predicted contact map from CMAPpro [2] is used to decide whether non-adjacent segments have a sufficient number of residue-residue contacts to be considered a single domain. The latter strategy is currently under development.

3. Results

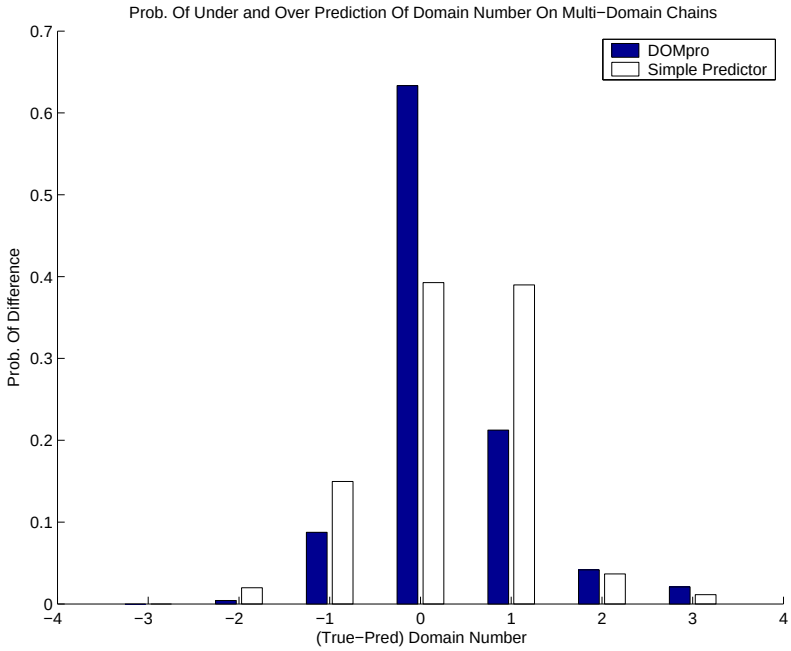


Fig. 7. Frequency of under and over prediction of the number of domains by DOMpro and a naive predictor

The first step in evaluating a domain predictor is to compare the predicted number of domains to the true number of domains. DOMpro correctly predicted the number of domains for 69% of the combined dataset of single and multi-domain proteins. 79% of the single domain proteins were correctly predicted as having no domain boundaries. The number of domains is correctly predicted for 43% of the multi-domain proteins. Figure 7 shows the relative frequency of under and over prediction of the number of domains by DOMpro in addition to a predictor based solely on the chain lengths. This simple predictor classifies a chain as having one domain if its length is less than 220 residues, two domains if its length is between 220 and 400 residues, three domains if its length is between 400 and 600 residues and four domains if its length is greater than 600 residues. These thresholds come from statistics on the number of residues per domain.

DOMpro is able to correctly predict both the domain number and domain boundary location for 20% of the multi-domain chains. For the evaluation of multi-domain chains, we consider that a domain boundary has been correctly identified if the predicted domain boundary is within 20 residues of the true domain boundary, as annotated in the CATH database.

The comparison of domain predictors is complicated by the existence of several domain datasets/databases which sometimes conflict with each other. Thus, the performance of a predictor on a dataset other than its training dataset is limited by the percentage of agreement between the training and testing datasets. With this caveat in mind, we observe that DOMpro

is able to correctly predict 25% of the two domain proteins in our dataset derived from CATH. This is in comparison to CHOPnet [13] which achieves 19% accuracy on a different dataset derived from CATH and SCOP. CHOPnet is reported to correctly predict the number of domains for 38% of the proteins, versus 43% for DOMpro. For single domain proteins, the performance of SnapDRAGON [6] is 44%, the performance of CHOPnet is 70%, and the performance of DOMpro is 79%. Thus, within the limitations of such comparisons, the performance of DOMpro is comparable to that of current *ab inito* domain predictors.

4. Conclusions

We have created DOMpro, an *ab initio* predictor of protein domains using recursive neural networks that leverages evolutionary information in the form of profiles and predicted secondary structure and relative solvent accessibility. DOMpro raw output is filtered in order to produce the final domain segmentation and assignment. Our analysis shows that DOMpro achieves a level of performances that is better or comparable to the level of current *ab initio* domain boundary predictors.

Domain prediction, however, remains a challenging problem. A 25% correct performance on prediction of two-domain proteins is encouraging but not sufficient and clearly there is room for improvement. We are currently adding a module to DOMpro to use homology for domain assignment for proteins that are homologous to known structures in the PDB and CATH databases. We are also training ensembles of predictors, although in preliminary experiments we did not see much improvement from using ensembles. In addition, we are focusing on the prediction/classification of discontinuous domains. To overcome the current limitations of DOMpro and the naive assignment of domain numbers, we are experimenting with the use of predicted contact maps, as well as domain length statistics, in deciding domain boundaries and whether or not two non-adjacent domain segments should be joined. The contact maps are predicted using 2D-RNNs [20,2]. The basic idea is that domains should be associated with a higher raltive density of contacts. Likewise, two discontinuous segments with the proper length statistics and with a sufficient number of inter-segment residue-residue contacts might be predicted as belonging to the same domain.

Acknowledgment

Work supported by the Institute for Genomics and Bioinformatics at UCI, a Laurel Wilkening Faculty Innovation award, an NIH Biomedical Informatics Training grant (LM-07443-01), an NSF MRI grant (EIA-0321390), a Sun Microsystems award, a grant from the University of California Systemwide Biotechnology Research and Education Program (UC BREP) to PB.

References

1. Altschul S, Madden T, Schaffer A, Zhang J, Zhang Z, Miller W, Lipman D, Gapped blast and psi-blast: a new generation of protein database search programs, *Nucleic Acids Res*, **25**(17), 3389–3402, 1997.
2. Baldi P, Pollastri G, The principled design of large-scale recursive neural network architectures-DAG-RNNs and the protein structure prediction problem, *Journal of Machine Learning Research*, **4**, 575–602, 2003.
3. Bengio Y, Fransconi P, Input-output HMM's for sequence processing, *IEEE Transactions on Neural Networks*, **7**(5), 1231–1249, 1996.
4. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne P, The protein data bank, *Nucleic Acids Research*, **28**, 235–242, 2000.
5. Chivian D, Kim D, Malmstro L, Bradley P, Robertson R, Murphy P, Strauss C, Bonneau R, Rohl C, Baker D, Automated prediction of casp-5 structures using the robetta server, *Proteins*, **53**(S6), 524–533, 2003.
6. George R, Heringa J, Snapdragon: a method to delineate protein structural domains from sequence data, *J. Mol. Biol.*, **316**, 839–851, 2002.

7. Holm L, Sander C, Parser for protein folding units, *Proteins Struct. Funct. Genet.*, **19**, 256–268, 1994.
8. Holm L, Sander C, Dictionary of recurrent domains in protein structures, *Proteins*, **33**, 88–89, 1998a.
9. Holm L, Sander C, Touring protein fold space with dali/fssp, *Nucl. Acids Res.*, **26**, 316–319, 1998b.
10. Jones DT, Protein secondary structure prediction based on position-specific scoring matrices, *J. Mol. Biol.*, **292**, 195–202.
11. Kabsch W, Sander C, Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, *Biopolymers*, **22**, 2577–2637, 1983.
12. Levitt M, Chothia C, Structural patterns in globular proteins, *Nature*, **261**(5561), 552–558, 1976.
13. Liu J, Rost B, Sequence-based prediction of protein domains, *Nucl. Acids Res.*, **32**(12), 3522–3530, 2004.
14. Marchler-Bauer A, Anderson JB, DeWeese-Scott C, Fedorova ND, Geer LY, He S, Hurwitz DI, Jackson JD, Jacobs AR, Lanczycki CJ, Liebert CA, Liu C, Madej T, Marchler GH, Mazumder R, Nikolskaya AN, Panchenko AR, Rao BS, Shoemaker BA, Simonyan V, Song JS, Thiessen PA, Vasudevan S, Wang Y, Yamashita RA, Yin JJ, Bryant SH, CDD: a curated Entrez database of conserved domain alignments, *Nucl. Acids Res.*, **31**(1), 383–387, 2003.
15. Marsden R, McGuffin L, Jones D, Rapid protein domain assignment from amino acid sequence using predicted secondary structure, *Protein Science*, **11**, 2814–2824, 2002.
16. Mika S, Rost B, UniqueProt: creating representative protein sequence sets, *Nucleic Acids Res.*, **31**(13), 3789–3791, 2003.
17. Murzin A, Brenner S, Hubbard T, Chothia C, Scop: a structural classification of proteins database for the investigation of sequences and structures, *J. Mol. Biol.*, **247**, 536–540, 1995.
18. Nagarajan N, Yona G, Automatic prediction of protein domains from sequence information using a hybrid learning system, *Bioinformatics*, **20**(9), 1335–1360, 2004.
19. Orengo C, Bray J, Buchan D, Harrison A, Lee D, Perl F, Sillitoe I, Todd A, Thornton J, The cath protein family database: a resource for structural and functional annotation of genomes, *Proteomics*, **2**, 11–21, 2002.
20. Pollastri G, Baldi P, Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners, *Bioinformatics*, **18 Supplement 1**, S62–S70, Proceeding of the ISMB 2002 Conference.
21. Pollastri G, Przybylski D, Rost B, Baldi P, Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles, *Proteins*, **47**, 228–235, 2001.
22. Pollastri G, Baldi P, Fariselli P, Casadio R, Prediction of coordination number and relative solvent accessibility in protiens, *Proteins*, **47**, 142–153, 2002.
23. Przybylski D, Rost B, Alignments grow, secondary structure prediction improves, *Proteins*, **46**, 195–205, 2002.
24. Wheelan SJ, Marchler-Bauer A, Bryant SH, Domain size distributions can predict domain boundaries, *Bioinformatics*, **16**(7), 613–618, 2000.