Striped sheets and protein contact prediction

Robert M. MacCallum
Stockholm Bioinformatics Center, Stockholm University, 106 91 Stockholm, Sweden
maccallr@sbc.su.se

Accepted for oral presentation at ISMB/ECCB 2004
To appear in a special issue of *Bioinformatics* (OUP)

Abstract

Current approaches to contact map prediction in proteins have focused on amino acid conservation and patterns of mutation at sequentially distant positions. This sequence information is poorly understood and very little progress has been made in this area during recent years.

In this study, an observation of "striped" sequence patterns across β-sheets prompted the development of a new type of contact map predictor. Computer program code was evolved with an evolutionary algorithm (genetic programming) to select residues and residue pairs likely to make contacts based solely on local sequence patterns extracted with the help of self organising maps. The mean prediction accuracy is 27% on a validation set of 156 domains up to 400 residues in length, where contacts are separated by at least 8 residues and length/10 pairs are predicted. The retrospective accuracy on a set of 15 CASP5 targets is 27% and 14% for length/10 and length/2 predicted pairs respectively (both using a minimum residue separation of 24). This compares favourably to the equivalent 21% and 13% obtained for the best automated contact prediction methods at CASP5. The results suggest that protein architectures impose regularities in local sequence environments. Other sources of information, such as correlated/compensatory mutations, may further improve accuracy.

A web-based prediction service is available at
http://www.sbc.su.se/~maccallr/contactmaps

Introduction

Protein structure prediction by simulated folding is difficult due to the huge search space and the use of energy functions that are inaccurate and computationally expensive. Because the three-dimensional structure of a protein may be reconstructed from inter-residue contacts, contact prediction offers a possible shortcut to predict protein structure. Contacts are defined as pairs of residues in a folded protein which are close in space, according to some predefined threshold. On a trivial level, pairs of hydrophobic residues are more likely to be in contact than other pairs of amino acid types, but even statistical pair potentials (Sippl, 1990) do not produce sufficiently specific contact predictions. More specific information appears to come from neighbouring residues and patterns of mutation, conservation and predicted secondary structure, all obtained from multiple sequence alignments of family members (Gobel et al., 1994; Taylor and Hatrick, 1994; Lund et al., 1997; Olmea et al., 1999; Fariselli et al., 2001; Fukami-Kobayashi et al., 2002; Pollastri and Baldi, 2002; Zhao and Karypis, 2003). Typically, neural networks or support vector machines are used to predict contacts or distance maps from these multiple sources information. From a theoretical point of view, the idea that compensatory (or correlated) mutations at two positions are indicative of physical interaction is attractive (Gobel et al., 1994). However, a recent study (Zhao and Karypis, 2003) and many years of evaluation of contact prediction at CASP (Lattman, 1995) have shown that this information is very weak. While more sophisticated methods exist to detect correlated mutational behaviour (Fukami-Kobayashi et al., 2002), they have yet to be tested in automated contact prediction. It may be the case that some other source of information must be exploited to improve contact prediction algorithms.

In this paper I describe a new technique for viewing amino acid sequence profiles on 3D structures. Using this, it is striking that in many proteins there are pairs of neighbouring strands with similar sequence patterns. This prompted the development of a new contact prediction algorithm, which was generated automatically by an evolutionary machine learning approach called genetic programming. The performance of this predictor, which uses only sequence profile and residue separation information as input, is approximately equal to or better than existing automated contact predictors which use more sources of information.

Datasets

SCOP domain sequences

A set of 2036 protein domains with pairwise sequence identity not more than 10% was taken from the ASTRAL subsets of SCOP release 1.55 (Brenner et al., 2000). These were split randomly into two sets: T (1573 domains) and V (463 domains), such that no pair of sequences belonging to different sets share the same SCOP superfamily. Subsets of T and V containing just one representative per superfamily were produced as follows: $T \rightarrow$ *training* (451 domains) and *testing* (227 domains) sets, and $V \rightarrow$ *validation* set (170 domains).

During training and evaluation of the contact predictors, only domains with length $L \leq 400$ are used. Also, a small number of domains with residue numbering or PDB format problems are discarded. As a result, the training, testing and validation sets effectively contain 430, 209 and 156 domains each.

CASP5 targets

For the retrospective CASP5 analysis, the same 15 "sequence-unique" targets from the EVA/CAFASP-3 evaluation (Eyrich et al., 2003) were used. These have no obvious homology to proteins in the Protein Data Bank at the time of the CASP5 experiment (May 2002) and therefore also neither to release 1.55 of SCOP (March 2001) which has been used for training and validation in this work. The targets range in size from 111 to 414 residues and are: T0138, T0146, T0147, T0148, T0149, T0156, T0157, T0159, T0161, T0172, T0173, T0174, T0181, T0187, T0193.

Sequence profile visualisation in 3D

Background

PSI-BLAST.

PSI-BLAST (Altschul et al., 1997) is a widely used tool for sensitive protein sequence database searching. Like a number of other search techniques it makes use of sequence profile information. Sequence profiles are basically a summary of the amino acid types present at each sequence position (column) in a multiple alignment of sequence family members, and may be calculated as frequencies or log-odds ratios, for example. During the evolution of a protein family, both structural and functional constraints influence the types of amino acids allowed at each position in the protein. A large part of the structural signal in sequence profiles is related to secondary structure.

The mapping between sequence and structure has been explored extensively in the literature, and most notably in the work relating to the I-sites library and HMMSTR local structure prediction tool (Han and Baker, 1996; Bystroff and Baker, 1998; Bystroff et al., 2000). It has been shown that sequence-structure correlations exist for different categories of helix, strand and turn, and also for supersecondary structure motifs. Although this area has been well studied, I was curious to try a different approach to local sequence profile clustering. Sequence profiles contain a lot of information. There are usually 20 or 21 values per sequence position, and so a window of 15 residues, for example, is represented by $15 \times 20 = 300$ values. In the next section, a technique which can help make some visual sense of this data is described.

Self organising maps.

Visualisation of high-dimensional data is always problematic. Many techniques exist to reduce the information into an observable number of dimensions (typically 2 or 3). These include principal components analysis, multidimensional scaling, singular value decomposition, and a host of clustering techniques, of which some are deterministic and others not. One common non-deterministic approach is known as k-means clustering. Here a predetermined number (k) of cluster centroids (means) are initialised randomly, then each data point is assigned to the nearest cluster centroid, then the centroids are recalculated, and the process continues until convergence. The self organising map (SOM) of Kohonen and Makisara (1989) is similar to the k-means algorithm, except that the reference vectors (equivalent to the centroids) are arranged on a grid and are somewhat connected. The standard SOM algorithm is outlined below, assuming d-dimensional "input" vectors:

- **initialisation:** create a 2D grid of d-dimensional vectors, v, with random starting values
- **training:** for each of E epochs:

```
1. for each data point $x$:
    (a) find the closest grid vector, $v_{win}$, to point $x$ according to some distance measure (e.g. Euclidean)
    (b) update $v_{win}$ towards $x$ by a small amount
            $v_{win} = v_{win} + \alpha(x - v_{win})$
    (c) update neighbours of $v_{win}$ within a certain radius $r$ in the same way, but by a smaller amount
2. reduce radius $r$ and training rate $\alpha$
```

- **application:** any data point x can be assigned to a "winning" grid vector, v_{win}

After training a SOM, any d-dimensional vector can be mapped to a position on the grid. The result is that data points which are close in the input space are mapped to the same or neighbouring grid nodes wherever possible. Thus, convoluted multi-dimensional data may be "flattened" onto a 2D grid with the maintenance of local (and to some extent, global) relationships. The SOM also gives more space/priority in the map to the more densely populated areas of the input space, and so also operates as a type of noise filter.

In this work, a variant of the standard SOM algorithm has been used. Firstly a 3D output grid was used. Secondly the neighbourhood function was square/cubic (normally it is circular) for implementation and speed reasons.

Mapping windows of sequence profiles

The amino acid sequences of 1573 protein domains of known structure (set T, described above) were extracted from their 3D coordinate files. Using scripts from the PSIPRED 2.3 package (Jones, 1999), PSI-BLAST searches were performed with each sequence against a set of non-redundant protein sequences (NCBI's "nr" data from 22 July 2002) and sequence profile matrices were generated. These matrices have 21 rows and L columns, where L is the length of the protein in residues. The total number of columns in this data is 262,503.

A series of $6 \times 6 \times 6$ SOMs were trained using 100,000 randomly selected $w \times 21$ submatrices (windows) extracted from the sequence profile matrices. When a window extends outside a matrix, it is padded with zeroes. Values for w between 1 and 15 were tried. The number of epochs was 6 and the SOM parameters were $\alpha = 0.1$ and $r = 3$.

After training, the SOMs can be used to map *all* overlapping sequence profile windows of a protein. The result is a list of winning map coordinates, a $L\times 3$ matrix of integer values ranging from 0 to 5 (the map is $6\times 6\times 6$, remember). For visualisation purposes, the integer triplets can be scaled and used as RGB (red, green, blue) triplets, i.e. colours. These colours can then be used for each residue in a 3D protein viewer. It is important to note that there is no correspondence between the colour schemes made with different SOMs (for example, using different window sizes or random seeds). The green region may mean "hydrophobic" in one map and "polar" in another, for example.

Example mappings

Figure 1 presents two protein domains with SOM-derived colour schemes. By comparing parts (a) and (b) it can be seen that that larger window sizes produce more distinctive colour patterns, particularly in the two central strands of the parallel These two strands have a very similar sequence of six colours (see Figure 1(b) and caption), which would occur by chance with a very low probability (something in the range of 10^{-12} to 10^{-5}). It is assumed that this striping is a consequence of the similarity of the structural environments through which the strands pass. The structural environments here are defined purely from sequence information like those of Han and Baker (1995) and are more fine-grained than those originally presented by Bowie et al. (1991). The strand pair relationships become more apparent as the window size is increased because the surrounding sequence information contains information about secondary structure which helps to pin down location within a protein fold (for example, N-terminus of exposed strand *vs.* C-terminus of buried strand). Colour sequence similarities between anti-parallel and "facing" strand pairs are also seen, as shown in Figure 1(c&d), although these seem to be less common.

Contact prediction

Overview of approach

If similar sequences of SOM-derived sequence profile classes/colours (now abbreviated to SDPCs) are indeed indicators of paired or nearby strands, then the information should be of use in contact prediction. One approach might be to use neural networks or similar tools to classify pairs of residues as contacting/non-contacting using SDPCs and sequence separation as input. However, an approach more closely related to non-linear regression is used here, with no particular justification except that the solution may have fewer parameters (to overfit) compared to neural networks. An outline of the prediction strategy applied to each protein sequence in this study is given below:

- produce a PSI-BLAST profile and perform SOM mappings (with pre-trained SOMs with different window sizes) to produce a set of $L\times 3$ SDPC matrices (`SDPCs`)
- for all residues i, apply the function: `residue_score(i, SDPCs)`
- define set R as the top scoring $L/5$ residues
- for all pairwise combinations i, j of residues R, (where $|i-j|\geq r$, and r is the minimum residue separation) apply the function: `pseudo_distance(i, j, SDPCs)`
- define the set P as the "closest" $L/10$ pairs
- classify the members of P as contacts and non-members as non-contacts

The functions `residue_score()` and `pseudo_distance()` will be evolved using genetic programming, as described below. First, I discuss their desired behaviour. The first "filter" function should preselect a set of residues that are likely to make contacts, such as hydrophobic and/or strand residues. The second "distance" function should attempt to automate some of the visual analysis described above, that is to return a small value for residue pairs belonging to parallel and anti-parallel strands with similar SDPC patterns. Also, if possible, the function should be able to identify contacts where other specific SDPC combinations are seen (perhaps involving helices), and also make use of $|i-j|$ (at large sequence separations, contacts are less likely). It might be possible to design such a function and then optimise the parameters by some means or other. As described below, however, genetic programming can "design" the function and optimise the parameters at the same time.

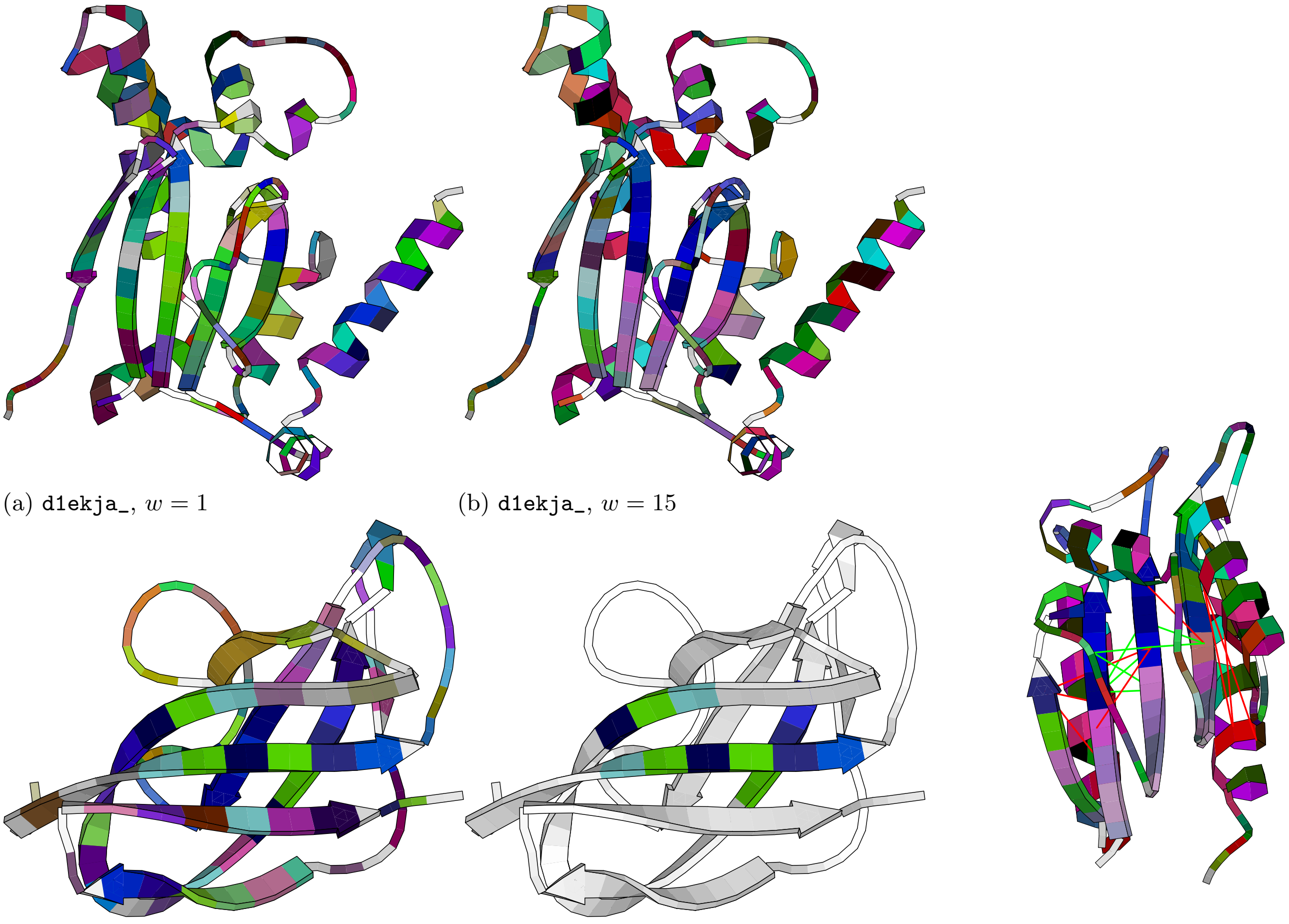
Genetic programming

Genetic programming (GP) is an evolutionary search/optimisation technique for generating computer program code to perform some particular task. Like the more familiar genetic algorithm (GA) the approach is inspired by nature and is based around a randomly initialised population of individuals. The individuals undergo selection on the basis of fitness with respect to the target behaviour, followed by reproduction, recombination and mutation. This is repeated until a satisfactory solution is produced.

In evolutionary algorithms, as in natural evolution, individuals are specified by genetic information (genotype), which is somehow transformed into a functional individual (phenotype). In a typical GA, the genotype is linear (like in biology), and the "genes" on this "chromosome" often correspond to the parameters of an optimisation problem. In GP, however, the genotype is often implemented with a tree-like chromosome, because this is a more convenient way to represent program code (see Figure 2). Manipulations are also often performed at the tree-level, and so a recombination event (also called crossover) would involve the swapping of subtrees between two individuals, and a macro-mutation might involve deleting an entire subtree, for example.

GP implementation

The open source GP system PerlGP (MacCallum, 2003) was used to produce the two functions discussed above. In



(a) `d1ekja_`, $w = 1$

(b) `d1ekja_`, $w = 15$

(c) `d1qhoa2`, $w = 15$

(d) `d1qhoa2`, $w = 15$

(e) `d1bvyf_`, $w = 15$

Figure 1: Molscript (Kraulis, 1991) cartoons of protein domains coloured using SOM-derived sequence profile information. In (a), a $w = 1$ SOM is used (single columns of profiles) to colour chain A of PDB entry 1EKJ (Kimber and Pai, 2000). Hydrophobic regions tend to have greenish colours but this is not very interesting. In (b) the same domain is shown with colours from a $w = 15$ SOM. Now there is a striking coincidence of the following colour sequence: purple, magenta, dark blue, “blue 1”, “blue 2”, “blue 3”, in the two central neighbouring strands. In (c) and (d), domain 2 (as defined by SCOP) from chain A of PDB entry 1QHO (Dauter et al., 1999) is shown with the same colour scheme as in (b). In (d), the anti-parallel and facing strands with similar colour patterns are highlighted. In (e), chain F of PDB entry 1BVY (Sevrioukova et al., 1999) is also shown with the $w = 15$ colouring scheme. Note again the purple to blue striping. The lines drawn between residues indicate predicted contacts (correct in green, incorrect in red).

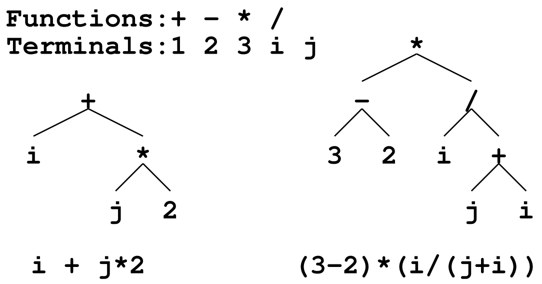


Figure 2: Trees are a natural representation for program code. In this trivial example, program fragments of any size can be built by choosing either a function or terminal node at each position in the tree. Two examples are given.

order to start evolving code with this Perl-based system, a small amount of Perl code must be written in the following three areas:

1. Data input

The training and testing data must be read into suitable data structures. For each protein in the dataset a series of four different SDPC matrices are generated with window sizes $w = 1, 5, 9, 15$. These $3 \times L$ matrices are stored in an array structure, which is referred to below as `@sdpc`. Each training instance must also have a desired or target output value associated with it: in this case the "real" $L \times L$ contact matrix calculated from the 3D protein structure coordinates. Here, a contact is defined as two residues whose carbon-β atoms (or carbon-α in the case of glycine) are closer than 8Å (in agreement with the CASP, EVA and Fariselli et al. (2001)).

2. Evolved code specification

The initial GP population will contain random individuals whose phenotype is a pair of function definitions. Although these functions are random they should be syntactically correct and should perform the types of operations required for the problem and not `system "rm -rf *"`, for example. This is made possible by following type-aware production rules or a *grammar*, as shown in Figure 3, to expand "skeleton" functions similar to the following:

```
sub pseudo_distance {
  my ($i, $j, @sdpc) = @_;
  # expand "Num" according to grammar
  return Num;
}
```

An infinite number of functions can be generated in this way (because their size is unbounded). The grammar produces arithmetic expressions of constants, input variables (`$i`, `$j`, and the contents of `@sdpc`), and calls to `eucwin(M,$i,$j,R,D)`. This accessory function returns the Euclidean distance between two windows (submatrices) in one of the SDPC matrices (`M`) centred around `$i` and `$j` with radius `R` (anti-parallel if `D` is negative). During evolution, the mutation and crossover operators also obey the grammar, so the functions continue to be valid.

```
Num ::= Num * Num | Num / Num |
        (Num + Num) | (Num - Num) |
        abs(Num) | sin(Num) | (Num)**2 |
        (Num > {Rconst} ? Num : Num) |
        Const | Rconst | Var |
        $sdpc[W][Var + Const][Y] |
        eucwin($sdpc[W], $i, $j, R, D)

Rconst ::= 0.0154 | 0.8661 | 0.2893 ...
Const  ::= -5 | -4 | -3 .. 5
W      ::= 0 | 1 | 2 | 3
Y      ::= 0 | 1 | 2
R      ::= 0 | 1 | 2 .. 5
D      ::= 1 | -1
Var    ::= $i | $j
```

Figure 3: Grammar, or production rules, for arithmetic expressions evolved in this work. It is shown in Backus-Naur form, with the grammatical categories as capitalised words (or single capital letters). To generate an expression, start with `Num` and replace it with one of the options (separated by `|`) on the right hand side of the `::=`. If the result contains a grammatical category, expand this in the same way. Continue until no more expansions are necessary. Some details, such as bound checking, are omitted for clarity.

Finally, for each protein in the training (or testing) set, a wrapper subroutine applies the two evolved functions to the sequence-derived data as outlined above (see "Overview of approach") to produce a predicted contact map. This is then passed to the fitness function.

3. Fitness evaluation

GP, like other evolutionary algorithms, requires a numerical measure of the *fitness* for the selection phase. The fitness measure used here is simply the contact prediction accuracy, which is defined as N_c/N_p, where N_c is the number of predicted contacts which correspond to true contacts in the target matrix, and $N_p = L/10$ (the number of predictions made). Local contacts, such as those made in turns and helices are not counted, since the residue separation cutoff used here is $|i - j| \geq 8$.

Evolving contact predictors

Twenty populations of 2000 individuals were run in parallel to evolve the `residue_score()` and `pseudo_distance()` functions for contact prediction. A standard PerlGP setup was used, including a tournament selection scheme where 50 individuals are chosen at random from the population, and the fittest 20 of these replace the least fit 20 individuals with their offspring. Occasional migration of individuals was allowed between populations, giving a total effective population of 40,000 individuals.

The pairwise application of the `pseudo_distance()` function is a rather time consuming step, even though it is only applied to $L/5$ residues (preselected by `residue_score()`). In order to reduce computation time

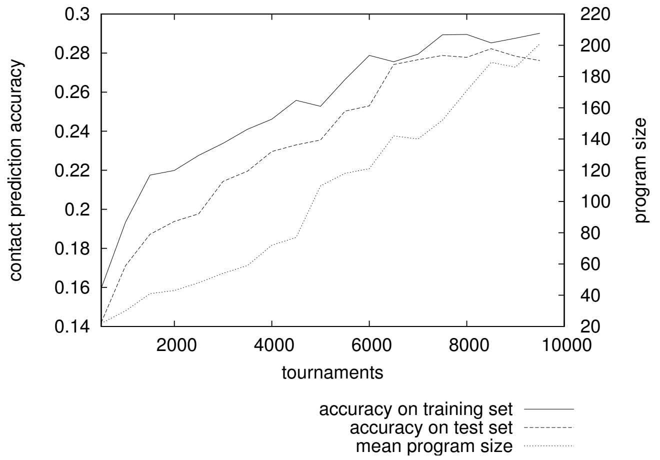


Figure 4: GP learning curve, showing mean accuracy and program size calculated over the 20 parallel populations. The ≈10,000 completed tournaments are equivalent to about 100 generations (where each individual in the population is evaluated once).

by a factor of around 4, fitness evaluations are performed on a random sample of just 100 domains from the training and testing sets. The subsets are re-sampled every 50 tournaments.

A steady increase in prediction accuracy was seen on both the training and testing data (see Figure 4) during just over 80 days of continuous processing. The test set fitness may have reached its peak value, or it could rise again – the only way to be sure is to run more tournaments. However, for now the mean contact prediction accuracy on the test set has reached an impressive 27.5%. There may be faster ways to reach the same result, of course. Note however that the actual application of the method takes just a few seconds in addition to a PSI-BLAST run. A hand-picked (and favourable) example of predicted contacts is shown in Figure 1(e).

Performance on validation set

Because sequence profiles from both the training *and* testing set domains were used to train the SOMs there could possibly be some over-estimation of the accuracy of the evolved contact predictors. The (so far untouched) validation set is therefore used in the analysis which follows.

One representative predictor was selected from the 20 populations totalling 40,000 individuals. Based on the accuracy over the combined training and testing sets, the best individual was identified from 50 randomly selected individuals from population 1. This individual (and no other) is then evaluated using the validation set. The results are presented in Table 1, and the behaviour of the new algorithm is probed by also calculating the accuracy on various subsets of the validation set.

The mean contact prediction accuracy over the entire validation set is 27.1% (using a minimum sequence separation of 8 residues, which is assumed unless otherwise stated). The SCOP class with the highest accuracy (39.2%) is "Small proteins". This class contains many non-standard proteins and peptides, such as snake and spider toxins. Another non-standard class, the "Membrane and cell surface proteins and peptides", is predicted very poorly. The sequence/structure environments in these proteins are very different from water-soluble globular proteins, so the poor performance is not surprising. Contacts in the remaining 127 "standard globular proteins" are predicted with 26.5% accuracy ("No Small or Membrane" in Table 1).

Table 1: Mean contact prediction accuracy (percent correct) on the validation set. For each protein domain of length L, $L/10$ predictions are made.

		min. separation		
Validation subset	n	8	16	24
Full set	156	27.1	24.6	20.6
SCOP class				
all-α	30	19.9	17.9	15.3
all-β	35	30.5	24.7	20.2
$\alpha + \beta$	40	25.9	24.2	21.2
α/β	18	31.5	36.7	34.9
Membrane assoc.	8	0.1	0.1	0.1
Multi-domain	4	23.5	18.6	18.4
Small proteins	21	39.2	32.9	22.0
$\alpha + \beta$ or α/β	58	27.6	28.1	25.4
Containing β	97	28.5	26.5	23.3
No Small or Membrane	127	26.5	24.5	21.4
Domain length, L				
$0 < L \leq 50$	15	34.4	22.2	13.3
$50 < L \leq 100$	51	30.7	30.3	25.4
$100 < L \leq 200$	53	23.7	21.3	18.5
$200 < L \leq 400$	37	24.2	22.4	20.0
$50 < L \leq 400$	141	26.4	24.8	21.4
Number of PSI-BLAST homologues				
≥ 15	140	27.3	25.5	21.3
≥ 50	108	29.4	27.7	22.9
≥ 100	78	29.8	27.8	23.5

In terms of secondary structural content, it seems that β-sheet containing domains are predicted best, with the highest accuracies calculated for all-β (30.5%) and mixed α/β (31.5%). However, one should be cautious because the number of domains is quite low, and t-tests generally fail to show significant differences between classes. Even if all β-containing classes are pooled and compared to all-α domains (28.5% *vs.* 19.9%), the difference is not (quite) significant at the 5% level. On the larger sample of the training and testing domains, this difference is clearly significant, however.

Accuracy appears to take a downward trend with increasing domain length, particularly above 100 residues. The largest 37 domains in the validation set are still predicted with a reasonable 24.2% accuracy, however.

At least two previous studies (Fariselli et al., 2001; Zhao and Karypis, 2003) have used datasets constructed with the requirement that each protein should have at least 15 protein family members for sequence profile generation. Here, no such limitation has been imposed, and when the number of homologues is taken into account during post-analysis (see Table 1), no large difference is seen when using a threshold of 15 family members. However, a reasonable increase (around 2%) is seen when 50 or more family members are required, and the incidence of do-

Table 2: Mean percent accuracy for 15 "sequence-unique" CASP5 targets at three coverage levels with residue separation ≥ 24.

	number of pairs predicted		
Method	$L/10$	$L/5$	$L/2$
Bystroff[a]	19	16	12
CMAPpro[b]	16	14	13
CORNET[c]	21	18	13
This work[d]	27	21	14

[a] Shao and Bystroff (2003)
[b] Pollastri and Baldi (2002)
[c] Fariselli et al. (2001)
[d] using a "frozen" sequence database contemporary to CASP5

mains with zero correctly predicted contacts appears to correlate inversely with the number of homologues (data not shown).

Retrospective performance on CASP5 targets

The data presented above cannot be easily compared with data from other groups because dataset construction and/or performance measures may differ. The CASP meetings (Lattman, 1995) and the continuous benchmarking experiment EVA (Eyrich et al., 2001) provide better conditions for direct comparison. The EVA team evaluated contact prediction servers at CASP5 and found accuracies between 12 and 15% for $2L$ predictions (see Figure 3 of Eyrich et al. (2003)). The authors kindly provided additional accuracy data (O. Graña, personal communication) for coverage levels $L/10$, $L/5$ and $L/2$. These are presented in Table 2 together with accuracies for the newly developed method on the same 15 targets with identical evaluation criteria. Although the sets are small, the higher accuracy with the new method is encouraging, particularly at lower coverage levels. Continuous evaluation at EVA has started, and in time will provide enough data for a more robust comparison.

Discussion

The coverage issue complicates the comparison of performance of this new method with others, which have largely been optimised for $L/2$ coverage. This method has been optimised for $L/10$ contacts, but it can also be evaluated using $L/2$ contacts. Using approximately the same criteria as Fariselli et al. (2001) ($L/2$ predictions, 8 residue separation, subset of validation set with > 15 homologues) the accuracy of this new approach is 19%. This compares to the published 21% of Fariselli et al., who used a neural network to classify contacts/non-contacts based on the following input information: profiles (3-residue window), correlated mutations, sequence conservation, $|i - j|$, and predicted secondary structure. The retrospective analysis on CASP5 targets presented above ranks these two methods in the opposite order. Therefore, the new method is competitive with (and perhaps better than) established techniques at lower coverage levels, and yet it uses only sequence profile information and $|i - j|$. The explanation seems to be that larger window sizes are used in this work. Why have others not used larger windows? In fact, some time ago Lund et al. (1997) found that windows of 18 residues were most effective for contact prediction. More recently, large profile windows may have been avoided because they introduce too many parameters into machine learning algorithms. Here the SOM has successfully reduced the amount of information to a manageable number of states. While secondary structure predictions also concisely summarise profile windows and are widely used in contact prediction (Fariselli et al., 2001; Pollastri and Baldi, 2002; Zhao and Karypis, 2003, for example), they are too coarse (usually just 3 states) to be used in isolation. It is interesting to note some convergence with the recent work of Shao and Bystroff (2003), where a finer subdivision of structural states ("I-sites" predicted super-secondary fragments) has been used in a statistics-based contact predictor.

A number of improvements to the current SOM- and GP-based approach can and should be made in the near future. For example, sequence conservation and correlated mutation information are not yet used, and intelligent post-processing (Fariselli et al., 2001; Shao and Bystroff, 2003) of predicted contact maps should also improve accuracy. More traditional statistical and machine learning approaches may also benefit from the SOM-processed profile information.

Does this study throw more light on the construction of protein folds? Strand pairing has been studied extensively (Hutchinson et al., 1998; Zhu and Braun, 1999; Mandel-Gutfreund et al., 2001; Steward and Thornton, 2002), however this study is the first to introduce a method for visualising local window-based sequence environments in 3D. With the SOM-derived colour schemes it becomes clear, particularly for parallel sheets, that there is often a synchronised transition of environments from the N to C-terminus of each strand. It is not yet clear if there is a link between these observations and protein folding, stability and evolution.

Acknowledgement

The author would like to thank Varun Aggarwal for his analysis of profile SOMs and secondary structure, Arne Elofsson for valuable suggestions and discussion, and the PDC for cluster management. SBC is funded by the Swedish Foundation for Strategic Research and computing facilities are funded by the Knut and Alice Wallenberg Foundation.

References

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J. H., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nuc. Ac. Res.*, 25:3389–3402.

Bowie, J. U., Luthy, R., and Eisenberg, D. (1991). A method to identify protein sequences that fold into a known 3-dimensional structure. *Science*, 253:164–170.

Brenner, S. E., Koehl, P., and Levitt, M. (2000). The ASTRAL compendium for protein structure and sequence analysis. *Nuc. Ac. Res.*, 28(1):254–256.

Bystroff, C. and Baker, D. (1998). Prediction of local structure in proteins using a library of sequence-structure motifs. *J. Mol. Biol.*, 281(3):565–577.

Bystroff, C., Thorsson, V., and Baker, D. (2000). HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins. *J. Mol. Biol.*, 301(1):173–190.

Dauter, Z., Dauter, M., Brzozowski, A. M., Christensen, S., Borchert, T. V., Beier, L., Wilson, K. S., and Davies, G. J. (1999). X-ray structure of Novamyl, the five-domain "maltogenic" alpha-amylase from Bacillus stearothermophilus: maltose and acarbose complexes at 1.7A resolution. *Biochemistry*, 38(26):8385–8392.

Eyrich, V. A., Marti-Renom, M. A., Przybylski, D., Madhusudhan, M. S., Fiser, A., Pazos, F., Valencia, A., Sali, A., and Rost, B. (2001). EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics*, 17(12):1242–1243.

Eyrich, V. A., Przybylski, D., Koh, I. Y. Y., Grana, O., Pazos, F., Valencia, A., and Rost, B. (2003). CAFASP3 in the spotlight of EVA. *Proteins: Struct., Funct., Genet.*, 53(Suppl 6):548–560.

Fariselli, P., Olmea, O., Valencia, A., and Casadio, R. (2001). Prediction of contact maps with neural networks and correlated mutations. *Protein Eng.*, 14(11):835–843.

Fukami-Kobayashi, K., Schreiber, D. R., and Benner, S. A. (2002). Detecting compensatory covariation signals in protein evolution using reconstructed ancestral sequences. *J. Mol. Biol.*, 319(3):729–743.

Gobel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins: Struct., Funct., Genet.*, 18:309–317.

Han, K. F. and Baker, D. (1995). Recurring local sequence motifs in proteins. *J. Mol. Biol.*, 251:176–187.

Han, K. F. and Baker, D. (1996). Global properties of the mapping between local amino-acid sequence and local-structure in proteins. *Proc. Natl. Acad. Sci. USA*, 93:5814–5818.

Hutchinson, E. G., Sessions, R. B., Thornton, J. M., and Woolfson, D. N. (1998). Determinants of strand register in antiparallel beta-sheets of proteins. *Prot. Sci.*, 7(11):2287–2300.

Jones, D. T. (1999). Protein secondary structure prediction based on position- specific scoring matrices. *J. Mol. Biol.*, 292:195–202.

Kimber, M. S. and Pai, E. F. (2000). The active site architecture of Pisum sativum beta-carbonic anhydrase is a mirror image of that of alpha-carbonic anhydrases. *EMBO J.*, 19(7):1407–1418.

Kohonen, T. and Makisara, K. (1989). The self-organizing feature maps. *Phys. Scripta*, 39:168–172.

Kraulis, P. J. (1991). Molscript — a program to produce both detailed and schematic plots of protein structures. *J. Appl. Cryst.*, 24:946–950.

Lattman, E. E. (1995). Protein-structure prediction — a special issue. *Proteins: Struct., Funct., Genet.*, 23:1.

Lund, O., Frimand, K., Gorodkin, J., Bohr, H., Bohr, J., Hansen, J., and Brunak, S. (1997). Protein distance constraints predicted by neural networks and probability density functions. *Protein Eng.*, 10(11):1241–1248.

MacCallum, R. M. (2003). Introducing a Perl Genetic Programming System: and Can Meta-evolution Solve the Bloat Problem? In *Genetic Programming, Proceedings of EuroGP'2003*, volume 2610 of *LNCS*, pages 369–378.

Mandel-Gutfreund, Y., Zaremba, S. M., and Gregoret, L. M. (2001). Contributions of residue pairing to beta-sheet formation: conservation and covariation of amino acid residue pairs on antiparallel beta-strands. *J. Mol. Biol.*, 305(5):1145–1159.

Olmea, O., Rost, B., and Valencia, A. (1999). Effective use of sequence correlation and conservation in fold recognition. *J. Mol. Biol.*, 293(5):1221–1239.

Pollastri, G. and Baldi, P. (2002). Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. *Bioinformatics*, 18(Suppl 1):62–70.

Sevrioukova, I. F., Li, H., Zhang, H., Peterson, J. A., and Poulos, T. L. (1999). Structure of a cytochrome P450-redox partner electron-transfer complex. *Proc. Natl. Acad. Sci. USA*, 96(5):1863–1868.

Shao, Y. and Bystroff, C. (2003). Predicting inter-residue contacts using templates and pathways. *Proteins: Struct., Funct., Genet.*, 53(Suppl 6):497–502.

Sippl, M. J. (1990). Calculation of conformational ensembles from potentials of mean force — an approach to the knowledge-based prediction of local structures in globular-proteins. *J. Mol. Biol.*, 213:859–883.

Steward, R. E. and Thornton, J. M. (2002). Prediction of strand pairing in antiparallel and parallel beta-sheets using information theory. *Proteins: Struct., Funct., Genet.*, 48(2):178–191.

Taylor, W. R. and Hatrick, K. (1994). Compensating changes in protein multiple sequence alignments. *Protein Eng.*, 7(3):341–348.

Zhao, Y. and Karypis, G. (2003). Prediction of contact maps using support vector machines. In *Proceedings of the 3rd IEEE International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 26–33.

Zhu, H. and Braun, W. (1999). Sequence specificity, statistical potentials, and three-dimensional structure prediction with self-correcting distance geometry calculations of beta-sheet formation in proteins. *Prot. Sci.*, 8(2):326–342.