Predicting protein structure classes from function predictions

I. Sommer\textsuperscript{1,}\textsuperscript{*}, J. Rahnenführer\textsuperscript{1,}\textsuperscript{†}, F.S. Domingues\textsuperscript{1}, U. de Lichtenberg\textsuperscript{2} and T. Lengauer\textsuperscript{1}

\textsuperscript{1}Department of Computational Biology and Applied Algorithmics, Max-Planck-Institute for Informatics, Stuhlsatzenhausweg 85, Saarbrücken D-66123, Germany and \textsuperscript{2}Center for Biological Sequence Analysis, BioCentrum-DTU, The Technical University of Denmark, Building 208, Lyngby DK-2800, Denmark

Received on August 8, 2003; revised on October 1, 2003; accepted on October 14, 2003
Advance Access publication January 29, 2004

ABSTRACT

Motivation: We introduce a new approach to using the information contained in sequence-to-function prediction data in order to recognize protein template classes, a critical step in predicting protein structure. The data on which our method is based comprise probabilities of functional categories; for given query sequences these probabilities are obtained by a neural net that has previously been trained on a variety of functionally important features. On a training set of sequences we assess the relevance of individual functional categories for identifying a given structural family. Using a combination of the most relevant categories, the likelihood of a query sequence to belong to a specific family can be estimated.

Results: The performance of the method is evaluated using cross-validation. For a fixed structural family and for every sequence, a score is calculated that measures the evidence for family membership. Even for structural families of small size, family members receive significantly higher scores. For some examples, we show that the relevant functional features identified by this method are biologically meaningful. The proposed approach can be used to improve existing sequence-to-structure prediction methods.

Availability: Matlab code is available on request from the authors. The data are available at http://www.mpi-sb.mpg.de/\textasciitilde sommer/Fun2Struc/

Contact: sommer@mpi-sb.mpg.de

INTRODUCTION

A prominent paradigm in protein function prediction is to first assign structure to sequence and then function to structure. Assigning structure to a sequence usually requires identifying a set of potential template structures that are then used to model the structure. Recently, several methods have become available for identifying function from sequence directly, including homology-based methods (Bork and Koonin, 1998), genomic context methods (Pellegrini, 1999; Marcotte, 2000; Huynen \textit{et al}., 2000), and methods that use relevant features derived from the sequences (Jensen \textit{et al}., 2002, 2003). For a recent review refer to Domingues and Lengauer (2003) and references therein. Here, we introduce the new approach to use the information contained in such sequence-to-function prediction data to detect classes of templates. Ultimately, these signals can be incorporated into existing sequence-to-structure prediction methods in order to improve accuracy. The outline of the paper is as follows:

- We describe a recently published method for producing functional classifications from sequence data. Given an amino acid sequence that method calculates probabilities for various functional properties.
- An intuitive and interpretable likelihood model is introduced. For every structural family, we first evaluate the usefulness of single functional categories for determining if a sequence belongs to the family. Then the most relevant categories are combined to an ensemble of classifiers.
- A cross-validation study is performed on a large set of sequences.
- The performance of the method is evaluated and the biological relevance of selected cases is discussed.

METHODS

We first briefly review the method to predict protein function. In the subsequent parts of this section, we describe the algorithmic details of our function-to-structure method.

Protein function prediction data

A successful method (ProtFun) for inferring human protein function entirely based on sequence information was recently presented by Jensen \textit{et al}. (2002). That method can be used
to assign proteins of unknown function to functional classes. In ProtFun, a variety of features is considered that can be calculated or predicted from the sequence. For a fixed functional category, a predictor using these features is constructed. A representative set of proteins is divided into training and test set and a feed-forward neural net is then trained to predict the functional class, based on a single input feature. Iteratively, larger networks are constructed comprising up to seven features, by choosing the best feature combinations in each step. For each class, the five best networks based on test set performance are combined into an ensemble. Applying this ensemble network to a query sequence then yields an annotation with a vector of probabilities for functional categories. We will refer to the elements of this vector as ProtFun values. ProtFun values are the output of the sequence-to-function method and the input of the function-to-structure method presented in this manuscript.

In our subsequent analysis, we use the 34 following functional categories as provided by the ProtFun server: amino acid biosynthesis, biosynthesis of cofactors, cell envelope, cellular processes, central intermediary metabolism, energy metabolism, fatty acid metabolism, purines and pyrimidines, regulatory functions, replication and transcription, translation, transport and binding [in total 12 Euclid functional categories, Andrade et al. (1999)], enzyme, non-enzyme, oxidoreductase, transferase, hydrolase, lyase, isomerase, ligase [in total six enzyme classes, Enzyme Nomenclature (1992)], signal transducer, receptor, hormone, structural protein, transporter, ion channel, voltage-gated ion channel, cation channel, transcription, transcription regulation, stress response, immune response, growth factor, metal ion transport [in total 14 Gene Ontology categories (The Gene Ontology Consortium, 2000)].

Likelihood model for assessing relevance of functional categories

Our goal is to introduce a method that fulfils three criteria: the method should use an objective likelihood model, should be robust enough to operate on sparse data, and should be easily interpretable.

We first describe our likelihood model. For a representative training set T of protein sequences (described below) we calculate all ProtFun values, i.e. probabilities for all functional categories. Consider now a fixed functional category and a fixed structural family. For each sequence of the training set we know its ProtFun value and whether it belongs to the structural family. Every pair of ProtFun values can be regarded as a candidate pair for the lower and upper boundary of an interval indicating family membership. We now determine an optimal interval of this type, in the sense that the family is statistically most over-represented, compared with a model based on random family membership. This means that we determine the interval, in which the observed number of family members, given the size of the interval, is most unlikely. We do a full search on those intervals whose boundaries are defined by two ProtFun values of sequences that both belong to the family of interest.

We denote the total number of sequences in the set T by N, the number of members of the selected family by n, and the number of all sequences with ProtFun values inside the interval by N_in. The probability p, that the ProtFun values of at least n_in out of the n family members lie within the selected interval at random, can be computed using a hypergeometric distribution as

\[ p = \sum_{i=n_{in}}^{\min(n, N_{in})} \frac{\binom{N_{in}}{i} \binom{N-n_{in}}{n-i}}{\binom{N}{n}}. \]  

We now determine the optimal interval by minimizing p with respect to all possible boundaries. For a given structural family and a given functional category, we thus divide all sequences into two groups, one with ProtFun values inside the interval boundaries and one with ProtFun values outside this most informative interval. In other words, we assume that there are informative regions of ProtFun values.

For an example of optimal intervals of ProtFun values (Fig. 1). Here, optimal intervals for the SCOP-superfamily ‘Metallothionein’ have been calculated. For better visibility, we replace ProtFun values by so-called relative ranks. Consider the set T of sequences. As a relative rank of a sequence we then denote the proportion of sequences in T that receive lower ProtFun values. A relative rank of 1 therefore corresponds to the highest score in the sequence set. For details on the selection of T, we refer to the subsection on the cross-validation study below. In Figure 1 one can see, that all family members receive low relative ranks for the functional categories 1, 2, 5, 13, 15, 17, 18 and high relative ranks for categories 14, 22, 23, 29. Such intervals thus contain important information about typical scores of family members.

We now estimate the odds ratio of an arbitrary sequence to belong or not to belong to a structural family. The likelihood ratio for the uniform distribution is given by the proportion of sequences inside and outside the interval, respectively, normalized by the relative breadth N_in/N of the interval. Let PF_s be the ProtFun value of the query sequence s and I_opt the optimal interval. The test statistic or score thus can be written as

\[ \text{score}(s) = \begin{cases} \frac{n_{in}}{n} \frac{N_{in}}{N}, & \text{if } PF_s \in I_{opt}, \\ \frac{n-n_{in}}{n} \frac{N-N_{in}}{N}, & \text{otherwise}. \end{cases} \]  

This basic Laplace model is intuitive and easy to interpret. Consider a functional category and a related ProtFun interval with a high density of members of a specific structural family. A new sequence of the same family should have an increased likelihood to have a ProtFun value in this interval. The optimal
Improvement through bias correction

Especially for structural families with only a small number of members, this basic algorithm is biased towards the training subset. Therefore, we have to perform a bias correction and calculate a larger interval.

Consider a uniform distribution on an interval \( (a, b) \) with unknown boundaries \( a < b \), and let \( x_1, \ldots, x_n, n > 1 \) be \( n \) realizations from this uniform distribution. Define \( \min = \min_{i=1,\ldots,n}(x_i) \) and \( \max = \max_{i=1,\ldots,n}(x_i) \). Then the minimum variance unbiased estimators for the interval boundaries \( a \) and \( b \) are given by \( \hat{a} = \min - (\max - \min)/(n - 1) \) and \( \hat{b} = \max + (\max - \min)/(n - 1) \), see for example Freund (1992).

Since usually \( ProtFun \) values are not uniformly distributed, a robust bias correction must be performed on the ranks of the observed values. Let \( r_l \) and \( r_u \) be the ranks of the sequences that belong to the lower and upper boundary of the optimal \( ProtFun \) interval \( I_{\text{opt}} = [PF_l, PF_u] \). We calculate

\[
\hat{r}_l = \max[1, r_l - [(r_u - r_l)/(n_{\text{in}} - 1)]] \quad \text{and} \quad \\
\hat{r}_u = \min[N, r_u + [(r_u - r_l)/(n_{\text{in}} - 1)]]
\]

and replace \( I_{\text{opt}} \) with \( \hat{I}_{\text{opt}} = [\hat{PF}_l, \hat{PF}_u] \), whose boundaries are the \( ProtFun \) values corresponding to the ranks \( \hat{r}_l \) and \( \hat{r}_u \).

We will refer to \( I_{\text{opt}} \) as the basic and to \( \hat{I}_{\text{opt}} \) as the unbiased interval.

Combination of single classifiers to an ensemble

For every single functional category this procedure yields a simple and intuitive classifier, which is only discriminative in cases with a dense cluster of \( ProtFun \) values. To improve the quality of the 34 basic classifiers, we combine them into one powerful ensemble classifier, which uses the incremental information given by its constituent components. We compare two methods for combining the 34 odds ratios obtained for a given query sequence. Both approaches are based on a pre-specified number \( c \) of categories. The 34 scores related to the sequence are sorted, and only the \( c \)-th largest of them are considered relevant. In one case, the \( c \)-th largest value itself serves as combined score, in the other case the product of all of the \( c \) largest odds ratios is calculated. We will call the resulting scores quantile score and product score.

The interpretation of the quantile score is that at least \( c \) functional categories are strong advocates for family membership. The \( c \)-th largest score gives a lower bound for the evidence from every single of the top \( c \) categories independently. Using the product score, the advocates are weighted according to their odds ratios. A very strong advocate can thus outweigh weaker advocates. This score is statistically optimal under the assumption of independence of the advocates. The true dependence structure of the advocates cannot be estimated reliably from the sparse data with only few data points for every structural class. Therefore, it is natural to combine them in a fashion that is as unbiased as possible. A good choice of \( c \) clearly depends on the total number of categories (advocates). In our cross-validation study (described in the following subsection) with 34 categories we
found that a small number such as $c = 5$ performs equally well for different variants of our algorithm. As can be seen from the biological relevance check in the last section, this number also leads to interpretable results.

We call the resulting overall procedure of interval selection, score calculation and combination weighted advocate voting.

**CROSS-VALIDATION STUDY**

For a reliable quality assessment of the proposed method we performed a cross-validation study on real data. A representative set of SCOP (Murzin et al., 1995) domains with <40% identity were downloaded from the ASTRAL server (Brenner et al., 2000; Chandonia et al., 2002), SCOP Version 1.59. We refer to the set of these domains’ sequences as $T$. Each sequence in $T$ is annotated with the SCOP classifications for its fold, superfamily and family. The algorithms described in the previous sections can be applied on any of these levels of hierarchy. Here, we focus on superfamily classification.

For all sequences in $T$, whether human or not human, output probabilities of ProtFun are obtained from a network that was trained on a specific sequence set $S$. We ensured that the similarity between sequences in $S$ and sequences in $T$ is low. We removed all sequences from $T$ that had a BLAST hit with an $E$-value smaller than 0.01 in $S$. This left 2027 sequences in our training set $T$. Table 1 and Figure 2 show distribution and histogram of the sizes of all structural superfamilies in $T$. Most superfamilies are rather small and therefore difficult to classify.

We applied the ProtFun method to all sequences in $T$ and obtained 34 ProtFun values for every sequence. Then $T$ was split in sets of size 10. Iteratively, one of these sets was removed from $T$ and optimal ProtFun values were estimated from the remaining 2017 sequences. For the remaining sequences, scores were calculated. We compared the basic algorithm and the unbiased extension both with product scores and with quantile scores (for their definitions see the preceding subsections). In all four cases, only the scores of the 10 sequences excluded from the training set were stored in a result matrix. This procedure corresponds to a 203-fold cross-validation.

For every fixed structural superfamily and fixed algorithm, the result matrix of the cross-validation contains scores for all sequences in $T$.

**RESULTS**

Suppose we apply one of the four versions of our algorithm to a sequence dataset $T$. As described in the previous paragraph about the cross-validation study, we obtain a result matrix of scores. Consider now an arbitrary but fixed superfamily. From the result matrix we can then retrieve exactly one score for every sequence in $T$.

<table>
<thead>
<tr>
<th>Size</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Superfamilies</td>
<td>391</td>
<td>146</td>
<td>68</td>
<td>37</td>
<td>17</td>
<td>21</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>Size</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td>Superfamilies</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Size</td>
<td>19</td>
<td>21</td>
<td>25</td>
<td>28</td>
<td>29</td>
<td>34</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>Superfamilies</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1. Distribution of superfamily sizes**

![Histogram of superfamily sizes.](image)

We want to check if members of the selected superfamily receive higher scores than non-members. This would indicate that useful information for superfamily classification has been learned from the input data. We use a Receiver Operating Characteristic (ROC) for the evaluation of our results. In a ROC plot, for various (score-) cut-offs the true positive rate (or sensitivity) is plotted against the false positive rate. True positive rate is the fraction of superfamily members with a score above the cut-off, and false positive rate is the fraction of non-members with a score above the cut-off. Consequently, ROC plots are an appropriate visualization of algorithm quality. Consider the point (0.9, 0.2) in a ROC plot. This means that 90% of superfamily members and 80% of non-members are classified correctly.

Since the distributions of scores vary strongly between superfamilies and we want to make statements across superfamilies (e.g. for bins of superfamily sizes) we compute relative ranks, which are ranks divided by the maximal rank. We analyze the distributions of relative ranks of superfamily members for the four versions of weighted advocate voting, discriminating between basic and unbiased intervals as well as between quantile scores and product scores. The number of relevant advocates is denoted by $c$, and the number of sequences in the training set $T$ is denoted by $N$. 
Fig. 3. AUC-values for all four versions of weighted advocate voting and for \(c = 1, \ldots, 34\), averaged over all sequences.

In a perfect scenario, superfamily members receive the ranks 1, \ldots, \(n\) out of the possible values 1, \ldots, \(N\). Then the false positive rate in the ROC plot is constantly 0 over the whole sensitivity range. In a scenario without learning where scores are assigned randomly, the false positive rate increases approximately linearly with the specificity. As a summarizing quality measure we use the AUC-value, the ‘area under the curve’ in the ROC plot. An AUC-value of 0 corresponds to perfect classification, whereas for random assignments an AUC-value around 0.5 can be expected.

Figures 3 and 4 show plots of AUC-values for all four algorithms and \(c \in \{1, \ldots, 34\}\), including all sequences (Fig. 3), and for sequences of large superfamilies (\(n \geq 14\)) only (Fig. 4). The product method clearly outperforms the quantile method and unbiased intervals perform slightly better than basic intervals. As expected, the combination unbiased product (UP) works best. An additional advantage of the UP method is that its performance barely depends on \(c\). For the quantile methods and superfamily sizes below 8, we observe a clear AUC-increase for many \(c\)-values. Although there are differences between methods, the major observation here is that for all parameters of the algorithm a considerable amount of learning takes place.

Figures 5 and 6 show ROC plots for UP with \(c = 5\) and \(c = 17\) advocates, respectively. We choose the value \(c = 5\) since this leads to a similar performance for all versions of the algorithm, and we choose the median \(c = 17 = 34/2\) since this leads to the best rankings for the UP method. Table 2 gives the exact values of false positive rates at sensitivities of 50 and 90%. For this analysis, the sequences are grouped according to the sizes of the superfamilies to which they belong. We choose bins of superfamily size 4–5 (54 members), 6–7 (39), 8–13 (24), 14–29 (16) and 30–38 (2).

Even for small superfamily sizes in the interval 4–29 scores of superfamily members are significantly higher than scores of non-members: For \(c = 17\), at sensitivity 50% we observe false positive rates of roughly 10%, and a sensitivity of 90% corresponds to false positive rates of roughly 50%. For the two largest superfamilies with size \(s > 30\) a sensitivity of 100% can be achieved with a false positive rate of 50%.

**BIOLOGICAL RELEVANCE**

For selected superfamilies we verify the relevance of the functional categories identified with the unbiased product weighted advocate voting algorithm. For each superfamily our method finds intervals for all functional categories that contain superproportionally many members of this superfamily (cf. Fig. 1). Every interval is labeled with a significance score, which is the value of the test statistic for a sequence with a ProtFun value inside the interval, see Equation (2).

A significance score for an interval stands for a dense cluster of ProtFun values for the corresponding functional category. This does not necessarily imply that all sequences of a structural superfamily fall into that cluster. For the classification algorithm this is not a disadvantage since the score of a sequence mostly depends on intervals which contain this sequence. On the other hand, it cannot be guaranteed that the most significant intervals belong to functional categories that are expected for this superfamily.
Function prediction helps structure prediction

For the superfamilies with the lowest AUC values (Table 3), we selected the most significant intervals and identified the corresponding functional categories. We then assessed the relevance of these selected functional categories in the corresponding structural superfamily by a two-step procedure. We first collected the SWISSPROT (Boeckmann et al., 2003) and (GOA) (Camon et al., 2003) annotations of the structures from the training set $T$ that belong to the structural superfamily. These annotations were then compared with the selected functional categories and matches were identified.

For the two smallest superfamilies, namely light-harvesting complex subunits and NADH oxidase/flavin reductase, the five most significant functional categories could not directly be linked to the biologically known function of proteins belonging to the respective superfamilies. For the other four superfamilies in this list there was considerable agreement between the significant functional categories and the functions observed in the members of these superfamilies. For the Metallothionein superfamily, we found agreement for the functional categories with the second and third highest significance score: transport-and-binding (score 31.6, high-rank values) and Nonenzyme (score 31.2, high-rank values). In the SCOP superfamily Invasin/intimin cell-adhesion fragments, we found matches for the categories with the highest, third and fourth significance score: Receptor (score 122.0, high-rank values), Replication-and-transcription (score 16.8, low-rank values), and Transport-and-binding (score 14.3, high-rank values). In the case of Six-hairpin glycosyltransferases, we found a match in Hydrolase, the category with the highest significance score (score 82.4, high-rank values).

The Membrane all-alpha SCOP superfamily ranks in the sixth position according to AUC values. It has 34 members and is the second largest superfamily. Here, the categories with

---

**Table 2.** False positive rates at sensitivities 50 and 90% for the unbiased product weighted advocate voting algorithm

<table>
<thead>
<tr>
<th>Size (superfamilies)</th>
<th>$c = 5$</th>
<th>$c = 17$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50%</td>
<td>90%</td>
</tr>
<tr>
<td>4–5 (54)</td>
<td>0.166</td>
<td>0.592</td>
</tr>
<tr>
<td>6–7 (39)</td>
<td>0.138</td>
<td>0.547</td>
</tr>
<tr>
<td>8–13 (24)</td>
<td>0.127</td>
<td>0.509</td>
</tr>
<tr>
<td>14–29 (16)</td>
<td>0.133</td>
<td>0.466</td>
</tr>
<tr>
<td>≥30 (2)</td>
<td>0.027</td>
<td>0.156</td>
</tr>
</tbody>
</table>

**Table 3.** Superfamilies with the lowest AUC values for unbiased product weighted advocate voting

<table>
<thead>
<tr>
<th>Superfamily</th>
<th>No. of members</th>
<th>AUC-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light-harvesting complex subunits</td>
<td>5</td>
<td>0.003</td>
</tr>
<tr>
<td>Metallothionein</td>
<td>6</td>
<td>0.017</td>
</tr>
<tr>
<td>NADH oxidase/flavin reductase</td>
<td>4</td>
<td>0.021</td>
</tr>
<tr>
<td>Invasin/intimin cell-adhesion frags</td>
<td>6</td>
<td>0.029</td>
</tr>
<tr>
<td>Six-hairpin glycosyltransferases</td>
<td>7</td>
<td>0.034</td>
</tr>
<tr>
<td>Membrane all-alpha</td>
<td>34</td>
<td>0.036</td>
</tr>
</tbody>
</table>

---

**Fig. 5.** ROC plots for unbiased product weighted advocate voting, for $c = 5$ advocates and with superfamily sizes split into bins.

**Fig. 6.** ROC plots for unbiased product weighted advocate voting, for $c = 17$ advocates and with superfamily sizes split into bins.
the highest significance scores were Transporter (score 42.7, high-rank values), Replication and transcription (score 33.8, low-rank values), Metal ion transport (score 29.1, high-rank values), Oxidoreductase (score 27.4, high-rank values) and Energy metabolism (score 26.4, high-rank values). All these significant functional categories match functions observed in members of the superfamily.

CONCLUSIONS

We propose to incorporate sequence-to-function relationships into sequence-to-structure prediction methods and present a weighted advocate voting algorithm. The advocates are functional categories whose importance depends on their relevance on a training set. Our basic underlying maximum-likelihood model provides high interpretability. In an extended version, we correct for the bias of the original algorithm.

Even for structural superfamilies with no more than five members the data contain strong signals. Scores of superfamily members are significantly increased. For large superfamily size, almost all superfamily members score above the median of all scores. Thus, a score below the median clearly indicates that the corresponding sequence does not belong to a given structural superfamily. This provides a direct exclusion criterion, which we plan to incorporate into an existing sequence-to-structure prediction method.

For the superfamilies with the best learning results, we examine the most significant functional categories and analyze their biological relevance. It is observed that these functional categories match in general the functions that are characteristic for the corresponding superfamily.

ACKNOWLEDGEMENTS

This work originated at the international conference and research center for computer science, Schloss Dagstuhl, where the authors met at Seminar 02471 on Computational Biology in November 2002. Financial support was provided by BMBF grants no. 01SF9984/3 (Helmholtz Network for Bioinformatics) (I.S.) and no. 031U117 (J.R.).

SUPPLEMENTARY DATA

For Supplementary Data please refer to Bioinformatics online.

REFERENCES


