JSSPrediction: a Framework to Predict Protein Secondary Structures Using Integration

L. Palopoli - DEIS, Università della Calabria, Italy - palopoli@deis.unical.it
S. E. Rombo - DIMET, Università “Mediterranea” di Reggio Calabria, Italy - rombo@unirc.it
G. Terracina - Dip. di Matematica, Università della Calabria, Italy - terracina@mat.unical.it
G. Tradigo - ICAR-CNR, Rende, Italy - gtradigo@si.deis.unical.it
P. Veltri - Università “Magna Græcia” di Catanzaro, Italy - veltri@unicz.it

Abstract

Identifying protein secondary structures is a difficult task. Recently, a lot of software tools for protein secondary structures prediction have been produced and made available on-line, mostly with good performances. However, prediction tools work correctly for families of proteins, such that users have to know which predictor to use for a given unknown protein. We propose a framework to improve secondary structure prediction by integrating results obtained from a set of available predictors. Our contribution consists in the definition of a two phase approach: (i) select a set of predictors which have good performances with the unknown protein family, and (ii) integrate the prediction results of the selected prediction tools. Experimental results are also reported.

1. Introduction

Proteins functionalities are strictly related to their amino acid sequences and to their spatial configuration. When a new protein is discovered, finding its spatial configuration is an important and yet not simple task. Protein structures are determined experimentally by x-ray crystallography or by nucleic magnetic resonance spectroscopy. Although accurate, such methods are time consuming while proteins discovering rate is much higher. Thus, recently, several secondary structure prediction software tools have been proposed and are available on-line [10, 1, 4, 8, 15], achieving good prediction accuracies. Nevertheless, improvements in the quality of prediction are constantly to be looked for. International challenges, such as the biannual competition CASP [3], have been instituted to encourage initiatives studying and designing precise synthetic predictions, witnessing the high interest of biomedical communities for bioinformatics tools. A problem is that most of such prediction tools have high accuracy only on specific groups of proteins. A challenging problem is therefore to devise a prediction method capable to achieve high levels of accuracy independently from the input protein group.

To improve the quality of prediction and to reduce the input dependency, recent methods based on joint use of available prediction tools have been proposed [4, 6, 11]. Similarly, the approach presented in this paper is also based on exploiting several existing prediction tools, available online. The difference with existing methods is that prediction results are integrated in order to obtain higher accuracy than using a single prediction tool. The integration consists in selecting the most accurate parts of each prediction result, recombining them in order to compose the final prediction for the input protein. However, since prediction tools depend on protein families, some of the prediction tools may return very low accuracy predictions. During integration, such predictions contribute negatively. In our approach we define a prediction selection phase to discard those predictors potentially giving negative contributions to the final prediction results. The presented ideas have been implemented in a prototype called JSSPrediction.

2 Related Work

A number of prediction methods have been presented in the last years and made available on-line; some are based on neural networks [9, 13, 12], others on sequence alignment [7] or fold recognition [5]. Most of such techniques exploit the input primary structure and other kinds of structural information to infer the correspondent secondary structure. The method presented here, uses available prediction tools. It first gives the amino acid sequence in input to several available prediction tools, then works selecting a subset of prediction tools that, supposedly, are able to provide posi-
Amino acids (as sequence of symbols representing conformation state of a predictor, and its associated predicted secondary structure is represented by another sequence where an amino acid residue can be associated to one of the three possible states: helix, strand or loop). In the secondary structure, an amino acid sequence, its secondary structure is represented by the corresponding amino acid in the input sequence. Figure 3 reports the sequence of amino acid (primary structure) used as input for prediction, and its associated predicted secondary structure as sequence of symbols representing conformation state of amino acids (H for helix, E for strand and L for loop).

Given an input amino acid sequence, the presented approach is based on two steps: (i) select a set of prediction tools having good performances with the unknown protein family, and (ii) integrate the prediction results of the selected prediction tools. To perform both steps some measure parameters are defined in the following.

3.1 Definitions

To measure the accuracy of a prediction, some parameters have been defined in the literature [14, 16]. Given a prediction tool and the amino acid sequence for a protein \( p \), the three-state prediction accuracy \( Q3 \) represents the percentage of secondary structure configurations (i.e., states) correctly predicted by the prediction tool. The per-segment accuracy \( SOV \) measures the percentage of segments of secondary structure correctly predicted, where a segment is a contiguous set of amino acids. \( Q3 \) and \( SOV \) can be evaluated once the real (observed) protein secondary structure is available.

Using such parameters, new ones are here defined in order to evaluate the accuracy of a prediction tool w.r.t. a set of proteins. In particular, given a secondary structure prediction tool \( T_i \), and a set \( P \) of \( m \) proteins whose observed secondary structure is known, the average per-segment accuracy coefficient \( SOV(i) \), is defined as follows:

\[
SOV(i) = \frac{\sum_{j=1}^{m} SOV(i,j)}{m} \times 100
\]  

where \( SOV(i,j) \) indicates the value of \( SOV \) corresponding to the prediction of \( T_i \) for the \( j \)-th protein in \( P \). \( SOV \) indicates the ability of a prediction tool to correctly predict entire sections of secondary structures. Such information is necessary to evaluate how much the “opinion” of such prediction tool is to be considered accurate, whenever a situation of disagreement among prediction tools occurs.

Given a set \( T \) of \( n \) prediction tools, and given a protein \( p \) with \( k \) amino acids, consensus parameter to measure the agreement among prediction tools in \( T \) while predicting the secondary structure of \( p \) is defined. In particular let \( T_i \in T \) be a prediction tool, and \( k_j \) the \( j \)-th amino acid in \( p \), the consensus percentage \( C(i,j) \) is defined as follows:

\[
C(i,j) = \frac{Nc(i,j)}{n} \times 100.
\]

where \( Nc(i,j) \) is the number of prediction tools in \( T \) that have predicted the same result as \( T_i \) for the \( j \)-th amino acid of \( p \).

The consensus percentage indicates how much a prediction tool agrees with the remaining \( n-1 \) ones in predicting a single amino acid state. Similarly, given a prediction tool \( T_i \) and a segment \( s_j \) in the predicted structure for \( p \), in order to evaluate the consensus of the prediction tool \( T_i \) with the remaining \( n-1 \) prediction tools in \( T \) w.r.t. the segment \( s_j \), the superposition mutual coefficient for segment \( SOV_{mutual} \) is defined as:

\[
SOV_{mutual}(i) = \frac{\sum_{l=1, l \neq i}^{n} SOV(T_l; T_i)}{n-1}
\]  

<table>
<thead>
<tr>
<th>name</th>
<th>Jpred</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary structure</td>
<td>AFDWNYTQAKLW - VSYRFQ</td>
</tr>
<tr>
<td>secondary structure</td>
<td>LLLEEEEEEELLHHHH - EEEELL</td>
</tr>
</tbody>
</table>

Figure 1. Primary and secondary structures.
where $SOV^{T,T_i}$ is analogous to $SOV$ with the difference that $SOV$ is evaluated by considering a predicted and an observed secondary structure, whereas $SOV^{T,T_i}$ measures the segment overlap existing between two predicted structures.

### 3.2 Team Selection

This section describes a procedure to define a set of prediction tools out of a set of available ones that, by working jointly, are able to produce more accurate predictions for the input protein. Such a set is named team in the rest of the paper. The team definition procedure is based on the following observation. If a group of prediction tools give high accuracy results working jointly on a family of known proteins, they shall behave analogously when applied on predicting an unknown protein of the same family.

Let $P$ be a set of $k$ proteins whose secondary structures are known and belonging to the same family of the unknown input protein $p$. Let $T$ be a set of prediction tools whose predictions are known for all the proteins in $P$. To generate a team of prediction tools $ST \subseteq T$, the less accurate prediction tools in $T$ for the proteins in $P$ have to be discarded. The selection algorithm works as follows. At the first stage $ST$ is set equal to $T$. The prediction tools in $ST$ are jointly used to predict proteins in $P$. The jointly use consists in using prediction tools as they were a single prediction tool, by integrating their contribution using the integration algorithm described in the next section. When $ST$ works as a single prediction tool, it is possible to evaluate for it the $SOV_{ST}$ parameter on the proteins $P$. At a first step, all prediction tools in $ST$ are ordered with respect to their $SOV_i$. Then, prediction tools are eliminated from $ST$, choosing the one with lowest $SOV_i$, and the remaining ones are jointly used to evaluate once again the predictions of proteins in $P$, and evaluating the new $SOV_{ST}$. If predictions of $P$ proteins are not improved (i.e., the new $SOV_{ST}$ is lower than the previous $SOV_{ST}$), last removed prediction tool is re-inserted in $ST$, otherwise (prediction improved, i.e. the new $SOV_{ST}$ is greater than the previous $SOV_{ST}$), the algorithm definitively discards the selected prediction tool and tries to improve $ST$ set by removing another prediction tool. The algorithm terminates when it has considered the elimination of each prediction tool from $T$, according to the described procedure.

The above described algorithm is reported in pseudo code in the following.

```plaintext
procedure Choose Team(T; P; ST)
begin
    ST = T;
    for each predictor $t_i \in T$ do
        compute the coefficient $SOV$ of $t_i$ w.r.t. $P$;
        order the prediction tools in $T$ w.r.t. $SOV$,
        from the less accurate predictor to the most accurate one;
        evaluate the coefficient $SOV$ of $ST$ w.r.t. $P$;
        take each predictor $t_i \in ST$ in turn according to the computed ordering, beginning from the less accurate one and do begin
            ST' = ST - $t_i$;
            compute the $SOV$ of $ST'$ w.r.t. $P$;
            if ($SOV$ of $ST'$ > $SOV$ of $ST$) then begin
                ST = ST';
                $SOV$ of $ST$ = $SOV$ of $ST'$;
            end;
        end;
        end;
end
```

### 3.3 Integration Algorithm

The integration algorithm has in input the amino acid sequence (primary structure) of a protein $p$ whose secondary structure is unknown, a set $ST$ of prediction tools, and a set $P$ of $m$ proteins whose structures are known (observed), and belonging to the same family. The prediction tools $ST$ are queried and their secondary structure results is obtained. The integration algorithm, for each amino acid of the target protein $p$, have to choose among the $n$ predicted states ($\alpha$-helix, $\beta$-strand and so on), composing the final result. If all predictors agree, the choice is obvious; in the opposite case, the choice is based on a voting matrix $M$ of size $m \times n$, of real number, where $m$ is the number of amino acids of $p$ and $n$ the number of used predictors. $M[i,j]$ contains a vote for the result of the $i$-th prediction on the $j$-th amino acid of $p$. The prediction tool with maximum vote in correspondence of a residue, is considered the more reliable. $M$ is defined by using the parameters defined in Section 3.1, evaluated running the prediction tools in $ST$ on a set $P$ of proteins whose secondary structure is known. The voting matrix $M$ is then defined as follows:

$$M[i, j] = SOV_{(i)} + C_{(i,j)} + SOV_{\text{mutual}(i)} \quad (4)$$

In particular, $SOV_{(i)}$ can be considered as a reliability score for the $i$-th prediction tool to predict the secondary structures of the proteins in $P$. $C_{(i,j)}$ represents a punctual agreement among the $i$-th prediction tool and the remaining ones in predicting the $j$-th amino acid of $p$, whereas $SOV_{\text{mutual}(i)}$ represents a structural agreement index comparing the $i$-th prediction w.r.t. the remaining ones. The integration algorithm is reported in the following.

```plaintext
procedure Integrates(p, T, M, s)
begin
    for each amino acid $j$ of $p$ do begin
        if all the prediction tools in $T$ agree on the amino acid $j$ then
            $s[j]$ = the $j$-th symbol in the prediction of a tool in $T$;
        else begin
            select $i$ such that $M[i, j]$ is the maximum of the column $j$ of $M$;
            $s[j]$ = the $j$-th symbol of the prediction of the $i$-th prediction tool;
        end;
    end;
end
```
The integration algorithm obtains a secondary structure prediction \( s \) for \( p \) as follows. For each amino acid \( j \) in \( p \), the prediction of the \( i \)-th tool is chosen, where \( i \) is obtained determining the maximum value \( M[i,j] \) for the column \( j \). Finally, the prediction \( s \) is the sequence obtained by concatenating the predictions chosen for each amino acid in \( p \).

4. JSSPrediction Tool

The approach presented above is part of JSSPrediction, a secondary structure prediction tool based on integration. JSSPrediction consists of three modules: (i) prediction tools querying and normalization; (ii) team selection and (iii) results integration. The first module is devoted to query available prediction tools by using amino acid sequences of both the input protein and the set of related (known) proteins. Such module is also devoted to map the prediction results obtained by different prediction tools, in a uniform formalism. The latter uses three code letters to represent a secondary structure, and collects results in an XML file used as input for the second and third module. While such a procedure has been partially implemented, second and third modules are fully implemented in Java, and have been used for experimental evaluations. In particular, the second module implements the algorithm of team selection described above, and the latter module implements the integration algorithm.

4.1 Experiments

JSSPPrediction prototype has been tested considering sets of related proteins, each containing at most 6 proteins whose structures are published on the PDB database [2]. Tests have been run by cross validating proteins of each input set, assuming each protein unknown in turn. Thus a total of 21 prediction experiments have been performed. In particular, two kinds of experimental evaluation have been performed. The first kind consisted of the integration algorithm validation by running the integration module on the available protein sets and using a team of prediction tools defined a-priori by observations related to biological properties of the input proteins and the corresponding family. In such a case, 21 different integration runs have reported that in 85.7% of cases, the predicted structure by using integration is improved w.r.t. using single prediction tools. The SOV parameter has been used to compare results. Such results demonstrate the effectiveness of the designed integration algorithm.

The second set of tests have been devoted to validate the team selection module. In this case, only in the 35% of cases the integration algorithm, used with a set of prediction tools chosen by the team selection module, improves the SOV accuracy. Nevertheless in 52% of cases the difference between the SOV obtained by using integration and the SOV of the best prediction tool remains within a 1.5 to 5% gap. Therefore, whereas in a significant percentage of the cases our technique strictly improves the prediction accuracy, in most of the cases returns at least as accurate results as the best predictors. It is anticipated that such results are going to be significantly further improved if the protein training sets used for team selection are significantly larger than those we have utilized the experiments accounted for here. Current work is focusing on this point.

5 Conclusions

This paper describes a method to improve the accuracy in protein secondary structure prediction, according to some precision parameters accepted and adopted in the literature. The framework, implemented in JSSPrediction tool, is based on (i) the selection of a team of prediction tools and (ii) the integration of their prediction results, to obtain a final, integrated prediction for an input protein. We are working in order to automatize query and normalization of available predictors, thus to test team selection module on larger data sets. Moreover, we are developing a web-based version of JSSPrediction tool.

References


