An information theoretic approach for improving data driven prediction of protein model quality

Alfonso Montuori*, Giovanni Raimondo, Eros Pasero

Department of Electronics, Politecnico di Torino, 10139 Torino, Italy

Abstract

We present the results of an information theory-based approach to select an optimal subset of features for the prediction of protein model quality. The optimal subset of features was calculated by means of a backward selection procedure. The performances of a probabilistic classifier modeled by means of a Kernel Probability Density Estimation method (KPDE) were compared with those of a feed-forward Artificial Neural Network (ANN) and a Support Vector Machine (SVM).

Keywords: Protein structure prediction; Protein model quality; Feature selection; Relative entropy; Statistical learning

1. Introduction

The current phase of genomics is to catalogue, characterize and comprehend the entire set of functional elements encoded in the human and other genomes. The compilation of this genome ‘parts list’ will be an immense challenge. Well-known classes of functional elements, such as protein 3D structures, still cannot be accurately predicted from sequence information alone because many features must be considered and it is not clear which features are most important and how they should best be combined [1]. On the other hand the experimental determination of the 3D structure of the proteins is currently a very laborious process. In some cases it can take years before the structure of a protein is determined, and in other cases, such as membrane proteins, the most commonly used methods are not always applicable [2].

The level of difficulty in predicting a protein structure is determined by the similarity of the protein sequence with that of a known protein structure. Scientists have classified protein structure prediction methods into three categories. From least difficult to most difficult, they are comparative modeling, fold recognition, and ab initio [3].

Several benchmarks of these methods have been developed, such as CASP [4], CAFASP [5], and LiveBench [6]. An important conclusion from these studies has been that for different targets, the best predictions are often made by different methods. This observation justifies the use of meta-predictors for the selection of a solution from a set of predictions coming from different servers. Basically a meta-predictor detects structural similarities

* Corresponding author. Tel.: +39 0 11 5644065; fax: +39 0 11 5644099.
E-mail addresses: alfonso.montuori@polito.it (A. Montuori), giovanni.raimondo@polito.it (G. Raimondo), eros.pasero@polito.it (E. Pasero).
between high scoring predictions selecting representatives of large clusters of models (consensus procedure). To fully benefit from the meta-predictors, it is important to understand what knowledge human experts use and what is not used by the servers (developed by them and others). There are two important contributions by the experts which should be automated and added to the consensus analysis: biological knowledge and structural verifications [7]. Due to the current limitations in computer-readable classifications of protein functions, biological knowledge is hard to automatize. Some attempts have been done by means of text-based bioinformatics approaches [8]. Text-based bioinformatics employs methods to retrieve or extract information from unstructured text for the purpose of structuring it into a biologically or medically relevant model.

In our work we focused on structural analysis of a protein model. A post-processing filter based on structural information might be very useful to deselect false positives if an appropriate set of structural features is selected. We know that as the number of features in a predictive task increases, the time requirements for an algorithm grow dramatically, sometimes exponentially. This problem is further exacerbated by the fact that many features in a learning task may either be irrelevant or redundant to other features with respect to predicting the class of an instance.

This paper presents the results of an information theory-based approach to select an optimal subset of structural features for the assessment of protein model accuracy. Atomic interactions, solvent accessibility, and secondary structure were considered and 109 structural features were defined. These three important categories of structural features have proven to include a subset of appropriate features for model quality prediction [9]. About 9000 instances of this set of features were calculated for as many protein models belonging to the Decoys ‘R’ US database [10]. The MaxSub score [11] was used to define 4 different classes of model accuracy and calculate the quality class of each model. These values were used to select an optimal subset of structural features by means of a backward feature selection algorithm based on the information theoretic measure of relative entropy (also known as Kullback–Leibler divergence).

Three different approaches were used to estimate the classification function which associates with each instance of the optimal subset of features the quality class of the related model.

First, a probabilistic classifier was used to model the classification function. The probabilistic classifier assigns the most probable quality class to an instance of the optimal subset of features, based on the probability distribution of the interdependencies among the class and the set of input features. The probability distribution was estimated with a Kernel Probability Density Estimation method (KPDE) [12]. Then a feed-forward Artificial Neural Network (ANN) and a Support Vector Machine (SVM) were trained to predict the MaxSub score of a protein model when an input vector corresponding to an instance of an optimal subset of features is presented. Finally the performances of the three approaches were compared.

2. Protein model quality

An important choice was to decide how to measure the quality of a protein model when the correct model is known. The assessment of predicted models is a very difficult task, and this has become an active subfield of research. The RMSD (Root Mean Square Deviation) computed over all atoms, is a very poor indicator of the quality of a model when only parts of the model are well-predicted. In fact the wrongly predicted regions could produce such a large RMSD that it is impossible to know if the model contains “well-predicted” parts at all. A second problem is that the RMSD strongly depends on the length of the protein. Several other traditional measures have similar problems [9,11].

For these reasons, the MaxSub score was used to measure the accuracy of a predicted protein model. Basically MaxSub computes a single scalar in the range of 0 to 1, which measures the similarity of a model to its corresponding experimental structure (0 for a completely wrong model, 1 for a perfect model). The scalar is a normalization of the size of the largest “well-predicted” subset (a well-predicted subset is a subset of model residues that superimpose upon their corresponding residues in the experimental structure within a distance threshold—usually 3.5 Å) and is computed using a variation of a formula suggested by Levitt and Gerstein [13].

The model accuracy was classified into 2, 3, and 4 levels based on the MaxSub score. For the 2-class partition the class with MaxSub < 0.1 and the class with MaxSub ≥ 0.1 were defined to indicate, respectively, an incorrect model and a correct model. The 3-class partition consisted of the following classes: MaxSub < 0.1, 0.1 ≤ MaxSub < 0.3, and MaxSub ≥ 0.3. Finally the 4-class partition was composed of the following classes: MaxSub < 0.1, 0.1 ≤ MaxSub < 0.3, 0.3 ≤ MaxSub < 0.6, and MaxSub ≥ 0.6.
3. Structural features

Protein structural information can be divided into different categories depending on the background principles they use. Many different categories of structural features for evaluating protein models have been considered [14]. In this work the main categories used by [9] were considered: atomic interactions, solvent accessibility, and secondary structure.

(A) Atomic interactions: to take into account the atomic interactions, the fraction of the noncovalently bonded atom–atom contacts between three (nonhydrogen) atom types (carbon, nitrogen, and oxygen/sulfur) in protein crystal structures were evaluated at different thresholds and sequence separations. As in [15] two atoms were defined to be “in contact” if the distance between the two atoms in space is less than some preset limit and the two atoms are not within the same residue or covalently bonded to each other. Three different distance thresholds were considered: 3.5, 5, and 7 Å. Furthermore five different classes of interaction were used to take into account the sequence separation s of the interacting atom pair: s ≥ 5, s ≥ 10, s ≥ 15, s ≥ 20, s ≥ 30. The total number of features in this category was 90.

(B) Solvent accessibility: the solvent-accessible surface area of a protein is the surface area that is accessible to a water molecule. Formally, the solvent accessibility surface is traced out by the probe sphere center as it rolls around the van der Waals surface of the protein [16]. The relative accessibility of each residue was calculated as the accessibility compared to the accessibility of that residue type in an extended ALA-x-ALA tripeptide (for amino acids) [17] using the program NACCESS [18]. The 20 amino acids were classified into 3 categories, based on their tendency to be either exposed or buried in the core of proteins [19]: surface amino acids, core amino acids, and intermediate amino acids. Six different classes of relative solvent accessibility were defined (<27%, <41.5%, <49%, <57%, <61.5%, and <81%) and the percentage of each category of amino acids falling in these six classes was calculated. The total number of features in this category was 18.

(C) Secondary structure: in the last years, the availability of large families of homologous sequences combined with sophisticated computing techniques such as neural networks revolutionized secondary structure prediction, leading to accuracies well in excess of 70%. Therefore only for very good protein models the accuracy in secondary structure assignments is comparable to what can be obtained by using the best programs for the prediction of the secondary structure from the sequence [20].

Based on these observations, it can be argued that a measure of similarity between predicted and model secondary structure could correlate with model quality. The similarity was measured as the percentage of residues which are classified as belonging to the same secondary class in predicted and model secondary structure. The DSSP program [21] was used to evaluate the model secondary structure. In order to predict the secondary structure from the sequence, the neural network-based program “SSPRO 4.03” [22] was used.

4. The feature selection algorithm

A feature selection method is needed when, as in our case, datasets with hundreds or thousands of features are available. In fact the many potential benefits of feature selection include: facilitating data visualization and understanding, reducing the measurement and storage requirements, reducing training and utilization times, defying the curse of dimensionality to improve prediction or classification performance [23].

An exhaustive search can conceivably be performed, if the number of features is not too large. But, the problem is known to be NP-hard and the search becomes quickly computationally intractable. A wide range of search strategies can be used, including best-first, branch-and-bound, simulated annealing, genetic algorithms, and greedy search strategies.

Depending on the way the searching phase is combined with the classification, there are three main classes of feature selection algorithms: filters, wrappers, and embedded.

A filter is defined as a feature selection algorithm using a performance metric based entirely on the training data, without reference to the classifier for which the features are to be selected. The name is derived from the way in which the features are filtered before the classification system is trained and tested. It is a general weakness of filter frameworks that feature subsets may rate highly, even when they are inappropriate or redundant to the classification algorithm being used, but it is not necessarily so. Efficient search strategies may be devised. Greedy search strategies,
such as forward selection and backward elimination, seem to be particularly computationally advantageous and robust against redundant features. In forward selection, features are progressively incorporated into larger and larger subsets, whereas in backward elimination one starts with the set of all features and progressively eliminates the least promising ones. Both methods yield nested subsets of variables.

Wrapper algorithms include the classification algorithm in the performance metric. The name is derived from the notion that the feature selection algorithm is inextricable from the end classification system, and is wrapped around it. Wrappers are often criticized because they seem to be a “brute force” method requiring massive amounts of computation.

Embedded methods perform feature selection in the process of training and are usually specific to given learning machines.

In this work a filter method was implemented because it was not known a priori which classification algorithm to use and it would have been computationally very expensive to compare the performances of wrappers based on different classification methods.

The Koller–Sahami algorithm [24] was used to select an optimal subset of features from the set of features described in Section 3. In the following the formalism of the authors will be used to describe the theoretical framework of the algorithm.

Let $F = (F_1, F_2, \ldots, F_N)$ be the set of structural features and let $Q = (Q_1, Q_2, \ldots, Q_M)$ be the set of protein quality classes. For each assignment of values $f = (f_1, f_2, \ldots, f_N)$ to $F$ we have a probability distribution $P(Q|F = f)$ on the different possible classes, $Q$. We want to select an optimal subset $G$ of $F$ which fully determines the appropriate classification. We can use a probability distribution to model the classification function. More precisely, for each assignment of values $g = (g_1, g_2, \ldots, g_P)$ to $G$ we have a probability distribution $P(Q|G = g)$ on the different possible classes, $Q$. Given an instance $f = (f_1, f_2, \ldots, f_N)$ of $F$, let $f_G$ be the projection of $f$ onto the variables in $G$. The goal of the Koller–Sahami algorithm is to select $G$ so that the probability distribution $P(Q|F = f)$ is as close as possible to the probability distribution $P(Q|G = f_G)$. To select $G$ the algorithm uses a backward elimination procedure, where at each state the feature $F_i$ which has the best Markov blanket approximation $M_i$ is eliminated. Formally, we say that a subset $M_i$ of $F$ which does not contain $F_i$ is a Markov blanket for $F_i$ if $F_i$ is conditionally independent of $F - M_i - \{F_i\}$ given $M_i$ [25]. If $M_i$ is a Markov blanket of $F_i$ then it is also the case that the classes in $Q$ are conditionally independent of the feature $F_i$ given $M_i$. The mean value of the relative entropy between the distributions $P(Q|M_i = f_{M_i}, F_i = f_i)$ and $P(Q|M_i = f_{M_i})$ is used to understand how close $M_i$ is to being a Markov blanket for $F_i$:

$$\delta_G(F_i|M_i) = \sum_{f_{M_i}, f_i} P(M_i = f_{M_i}, F_i = f_i)$$

$$\cdot \sum_{Q_i \in Q} P(Q_i|M_i = f_{M_i}, F_i = f_i) \cdot \log \frac{P(Q_i|M_i = f_{M_i}, F_i = f_i)}{P(Q_i|M_i = f_{M_i})}. \quad (4.1)$$

If $M_i$ is, in fact, a Markov blanket for $F_i$ then $P(Q|M_i = f_{M_i}, F_i = f_i)$ is equal to $P(Q|M_i = f_{M_i})$, in other words $Q$ is statistically independent of $F_i$ given $M_i$, and $\delta_G(F_i|M_i) = 0$. Hopefully, if $M_i$ is an approximate Markov Blanket, then $\delta_G(F_i|M_i)$ will still be low. The Koller–Sahami algorithm was modified in the way a candidate Markov blanket $M_i$ for the feature $F_i$ is selected. Instead of selecting a candidate Markov blanket $M_i$ of size $k$ for the features $F_i$ by using the set of the $k$ features most correlated to $F_i$, the $k$ features $F_j$ which minimize the mean value of relative entropy between the distributions $P(Q|F_i = f_i, F_j = f_j)$ and $P(Q|F_j = f_j)$ were selected.

The idea of using a Markov blanket to estimate the relative entropy could also be applied in the case of forward selection: we add to our current $G$ the feature $F_i$ that maximizes the relative entropy between $P(Q|G = g)$ and $P(Q|G = g, F_i = f_i)$. But, as Koller and Sahami remark in their paper [24], the forward selection procedure has some disadvantages with respect to the backward elimination procedure. In fact the goal of the selection procedure is to remain as close as possible to the correct conditional distribution $P(Q|F = f)$. If we use the forward procedure we begin with an empty set. So we start with the distribution $P(Q)$ given no feature and try to maximize the information gain by adding a new feature. There is no guarantee that taking a large step away from initial distribution actually gets us closer to the goal distribution. This latter argument can be illustrated by an example described by Guyon in her paper [23]. In that example, one variable separates the two classes better by itself than either of the other two taken alone and will therefore be selected first by forward selection. At the next step, when it is complemented by either of
Table 1  
Classification function performances

<table>
<thead>
<tr>
<th>Classification function</th>
<th>Number of target classes $M = 2$</th>
<th>Number of target classes $M = 3$</th>
<th>Number of target classes $M = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probabilistic classifier with relative-entropy-based feature selection</td>
<td>0.79</td>
<td>0.62</td>
<td>0.54</td>
</tr>
<tr>
<td>ANN with relative-entropy-based feature selection</td>
<td>0.80</td>
<td>0.63</td>
<td>0.55</td>
</tr>
<tr>
<td>SVM with relative-entropy-based feature selection</td>
<td>0.79</td>
<td>0.61</td>
<td>0.54</td>
</tr>
<tr>
<td>Probabilistic classifier with linear-correlation-based feature selection</td>
<td>0.73</td>
<td>0.56</td>
<td>0.49</td>
</tr>
</tbody>
</table>

the two other variables, the resulting class separation in two dimensions will not be as good as the one obtained jointly by the two variables that were discarded at the first step. A backward selection method may outsmart forward selection by eliminating at the first step the variable that by itself provides the best separation to retain the two variables that together perform best.

The computational complexity of this algorithm is exponential only in the size of the Markov blanket, which is small. For the above reason the probability distributions $P(Q|M_i = f_{M_i}, F_i = f_i)$ and $P(Q|M_i = f_{M_i})$ could be quickly estimated for each assignment of values $f_{M_i}$ and $f_i$ to $M_i$ and $F_i$. These probability distributions were estimated by calculating the instances $f = (f_1, f_2, \ldots, f_N)$ of $F$ and their corresponding quality class $Q_i$ on a database of about 8000 protein models representative of the Decoys ‘R’ US database. A KPDE method (Parzen window method) [26,27] was used to model the probability distributions. Decoys are computer generated conformations of protein sequences that possess some characteristics of native proteins, but are not biologically real. We used the decoys from the ‘4state-reduced’, the ‘fisa’, the ‘fisa-casp3’, and the ‘lmsd’ sets [10], taking care to select an equal number of ‘correct’ and ‘incorrect’ models.

5. The classification function

Popular predictors include decision trees, native Bayes, least-square linear predictors, probabilistic classifiers, neural networks, and support vector machines. In order to estimate the classification function three different approaches were used. First the classification function was modeled with the probability distribution $P(Q|G = g)$ estimated again by means of the Parzen window method on the protein models of the Decoys ‘R’ US database. The performances of this approach were evaluated on a separate set of 1000 protein models, representative of the Decoys ‘R’ US database. For each instance $g = (g_1, g_2, \ldots, g_P)$ calculated from a protein model of the test set, the most probable value of $Q$ calculated from $P(Q|G = g)$ was compared with the value of $Q$ calculated from the protein model and experimental coordinates with the MaxSub program. The fraction of correctly predicted protein model classes of quality are reported in Table 1 for the 2-class, 3-class, and 4-class partitions. The best performances were obtained with the following set of 6 features:

- Similarity between predicted and model secondary structure;
- Percentage of nitrogen–nitrogen contacts under the distance threshold of 3.5 Å with a sequence separation greater than 5.
- Percentage of carbon–oxygen/sulfur contacts under the distance threshold of 5 Å with a sequence separation greater than 5.
- Percentage of carbon–oxygen/sulfur contacts under the distance threshold of 3.5 Å with a sequence separation greater than 10.
- Percentage of oxygen/sulfur–nitrogen contacts under the distance threshold of 7.5 Å with a sequence separation greater than 30.
- Percentage of “surface amino acids” with surface accessibility less than 41.5%.

As expected, the similarity between predicted and model secondary structure and the solvent accessibility were selected. Since nitrogen and oxygen, present either in the backbone or in the side chains, are the polar component of amino acids, this could indicate that polar–polar contacts are an important factor for predicting model quality. However these results should be validated by repeating the feature selection procedure on a more comprehensive and representative database of protein models. The performances of this approach were compared to those achieved with a second KPDE-based probabilistic classifier, in which we used as input vector the 6 features showing the highest
linear correlation with the target. This subset was formed exclusively by carbon–carbon and nitrogen–oxygen contact percentages and performed worse than the subset selected with the relative entropy-based approach (see Table 1).

The performances of the KPDE-based probabilistic classifier decay for sizes of the optimal subset of features greater than 6 (see Fig. 1). This is mainly due to the lack of a sufficient number of protein models for the precise estimation of the probability distributions.

The same set of 8000 protein models of the Decoys ‘R’ US database was used to train a set of feed-forward ANN

The performances of the ANN and the SVM are shown in Table 1. The performances were evaluated on a separate set of 1000 protein models, representative of the Decoys ‘R’ US database (the same set used to test the probabilistic classifier), for different sizes of the input vector. The optimal subsets of features of different sizes selected by the feature selection algorithm in subsequent steps of the backward procedure were used as input vectors. The results in Table 1 refer to an ANN with 9 neurons in the hidden layer and an input vector of size 14 and to an SVM with an input vector of size 12, \( \epsilon = 0.001 \), and \( \sigma = 1 \). Contrary to the probabilistic classifier, both the ANN and the SVM did not show a decay of the performances when the size of the input vector increases (see Fig. 1). More precisely performance improves meaningfully when the size of the input increases from 1 to 10 and tends to flatten when the size of the input vector is greater than 10. On the other hand the probabilistic classifier performs better than the ANN and SVM when the size of the input vector is small.

The performance of the ANN depending on the number of neurons in the hidden layer shows a different behavior for the 3 different class partitions. While for the 2-class partitions the performance is nearly constant for a number of neurons greater than 3, the 3-class and the 4-class partitions undergo a greater improvement up to 10 neurons (see Figs. 2–4).

Different assignments for SVM parameters \( \epsilon, \sigma, \) and \( C \) were tried in order to find the configuration with the highest efficiency. As can be seen from Fig. 5, when we keep \( \epsilon \) and \( C \) constant (\( \epsilon = 0.001 \) and \( C = 1000 \), for example), the SVM results, for a large number of input features, depend on \( \sigma \) and reach a maximum when \( \sigma = 1 \), corresponding to an optimum trade-off between SVM generalization capability (large values of \( \sigma \)) and model accuracy with respect to
Fig. 2. Performances of the ANN classification model as a function of input vector size and number of hidden neurons, with a 2-class partition of the target values.

Fig. 3. Performances of the ANN classification model as a function of input vector size and number of hidden neurons, with a 3-class partition of the target values.

Fig. 4. Performances of the ANN classification model as a function of input vector size and number of hidden neurons, with a 4-class partition of the target values.

the training data (small values of $\sigma$). The value of $\sigma$ corresponding to this trade-off decreases to 0.1 for lower values of the input vector size, reflecting the fact that the generalization capability is less important when the training set
is more representative. If we keep $\sigma$ and $C$ constant ($\sigma = 1$ and $C = 1000$, for example), the best performances are achieved when $\varepsilon$ is close to 0 and the allowed training error is minimized (see Fig. 6). From this observation, by abductive reasoning we could conclude that the input noise level is low [34]. In accordance with such a behavior the performance of the network improves when the parameter $C$ increases from 1 to 1000 (see Fig. 7). Since the results tend to flatten for values of $C$ greater than 1000, the parameter $C$ was set equal to 1000.

From the computational complexity point of view, the ANN and the SVM provide a more concise model of the density, and so reduce the computational overheads associated with the Parzen window, which requires the storage of all the training data [35].

6. Conclusion

It has been shown that information-theoretic methods for feature selection can be efficiently used for the assessment of protein 3D models. Since our approach does not rely on the method used to predict the protein structure, it can be used in all the main tertiary structure prediction algorithms. In particular it can be combined with a traditional consensus procedure for the meta-prediction of protein structure. Future work will include the analysis of different methods to evaluate the quality of the models, including both alignment dependent and independent measures [36]. Furthermore a larger database with a set of protein models representative of the Protein Data Bank (PDB) of known protein structures [37] will be used. Finally the model assessment algorithm will be applied as the final step of a protein 3D structure meta-predictor and the performances will be compared with those of a traditional consensus procedure.
Fig. 7. Performances of the SVM classification model as a function of C (ε = 0.001 and σ = 1), with a 2-class partition of the target values.

Acknowledgments

We would like to thank the authors of the programs used for the scientific research related to this paper. We acknowledge Kabsch and C. Sander for the program DSSP, S. J. Hubbard and J. M. Thornton for the program NACCESS, N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer for the program MaxSub, J. Cheng, A. Randall, M. Sweredoski, G. Pollastri, and P. Baldi for the program “SSPRO 4.03”, S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy for the “SVM and Kernel Methods Matlab Toolbox”, and Luisa Pugliese, from S. A. F. A. N. Bioinformatics, who offered her external consultancy for this work.

A memento for Mario Costa: his precious work is fundamental in the mathematical aspects of this paper.

This work was partially supported by INFN STIPER project.

References

[34] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neurocomputing (2002) (special issue on SVM).