Predicting Protein Secondary Structure with Markov Models

Paul Fischer, Simon Larsen, and Claus Thomsen

Informatics and Mathematical Modelling
Technical University of Denmark
DK-2800 Kongens Lyngby

Abstract. The *primary structure* of a protein is the sequence of its amino acids. The *secondary structure* describes structural properties of the molecule such as which parts of it form sheets, helices or coils. Spacial and other properties are described by the higher order structures. The classification task we are considering here, is to predict the secondary structure from the primary one. To this end we train a Markov model on training data and then use it to classify parts of unknown protein sequences as sheets, helices or coils. We show how to exploit the directional information contained in the Markov model for this task. Classifications that are purely based on statistical models might not always be biologically meaningful. We present combinatorial methods to incorporate biological background knowledge to enhance the prediction performance.

1 Introduction

The primary structure of a DNA-sequence is given by the sequence of its amino-acids. The secondary structure is a classification of contiguous stretches of a DNA-molecule according to their conformation. We use a threefold classification, namely the conformation *helices*, *sheets*, and *coils*. Most databases contain a finer classification into 6 or more classes. We use the mapping from Garnier et al. (1996) and Kloczkowski et al. (2002) to reduce to the three aforementioned classes.

The task is to determine the secondary structure from the primary one. We use a supervised learning approach for this purpose. From a database one collects a number of DNA-sequences for which the classifications are known. On these a (statistical) model is trained and then used to assign classifications to new, unclassified protein sequences. There is a number of such classifiers which are based on different learning concepts. Some use statistical methods like, e.g., the GOR algorithm, Garnier et al. (1996) and Kloczkowski et al. (2002). GOR are the initials of the authors of the first version of this method: Garnier, Osguthorpe, and Robson. Other algorithms rely on neural networks like PHD, Rost and Sander (1993) and (1994). The acronym means “Profile network from HD”, where HD is the number plate code for Heidelberg, Germany, where the authors worked. Most of them incorporate biological background knowledge at some stage. For example a first classification given by a statistical model is then checked for biological plausibility and, if necessary, corrected.

We use a first order Markov model as classifier. This type of classifier has been successfully used before in a related setting, Brunnert et al. (2002). There, the order and length of the helix and sheet subsequences was given (but no information on the intermediate coil parts was known). Here, we investigate how this classifier performs without the additional information on order and length and how its performance can be improved. The aim is to push the basic statistical method to its limits before combining it with other techniques. We use the GOR algorithms as references. They have been re-implemented without the incorporation of background knowledge.

2 The method

Let Σ_a denote the alphabet for the 20 amino acids, and let $\Sigma_c = \{H, E, C\}$ denote the classification alphabet, where H denotes helix, E sheet, and C coil. In the following let $\mathbf{x} = x_1, \ldots, x_n$ be a protein sequence, where $x_i \in \Sigma_a$. Let $\|\mathbf{x}\|$ denote its length. Let $C = c_1, \ldots, c_n$ be the corresponding classification sequence, $c_i \in \Sigma_c$.

We shall use a first order Markov model for the prediction. The model uses a parameter ℓ, the *window size*. Such a model assigns probabilities p to subsequences of $\mathbf{x}$ of length l as follows:

$$p(x_i, \ldots, x_{i+\ell-1}) = p(x_i)\, p(x_{i+1} \mid x_i) \cdots p(x_{i+\ell-1} \mid x_{i+\ell-2}) \tag{1}$$

For the threefold classification task we have in mind, three such models are used, one for each of the classes $\{H, E, C\}$. The probability functions of the respective models are denoted by p_H, p_E, and p_C. The three models are trained by estimating their parameters of the kinds $p_X(\,\cdot\,)$ and $p_X(\,\cdot \mid \cdot\,)$ $X \in \{H, E, C\}$. Then they can be used for classification of new sequences as follows: One evaluates all three models and then assigns that class which corresponds to the model with highest probability: $\arg\max\{p_H, p_E, p_C\}$. The obvious problem with this approach is, that a Markov model assign probabilities to subsequences (windows) and not to individual amino acids. This might lead to conflicting predictions. If, for example, $E = \arg\max_X\{p_X(x_1, \ldots, x_{\ell-1})\}$ and $H = \arg\max_X\{p_X(x_2, \ldots, x_\ell)\}$, it is not clear which of the two classifications x_2 should get. We choose to assign the classification of a window to the first amino acid in that window. The estimation of the model parameters is then performed to support this choice. We denote this by using the term $p(i)$ for this, i.e.,

$$p_X(i) = p_X(x_i)\, p_X(x_{i+1} \mid x_i) \cdots p_X(x_{i+\ell-1} \mid x_{i+\ell-2}) \tag{2}$$

We investigated several modifications of the Markov model, some of which also differ in the training process. The basic training is conducted as follows.

The training data consists of N DNA-sequences $\mathbf{x}^{(j)}$ and the corresponding classification sequences $\mathbf{c}^{(j)}$, $j = 1, \ldots, N$. Now, three sets of subsequences are constructed, one for each of the three classes. Each DNA-sequence $\mathbf{x}^{(j)}$ is divided into maximal substrings according to the classification $\mathbf{c}^{(j)}$: Let $x_k^{(j)} x_{k+1}^{(j)} \cdots x_{k+\ell-1}^{(j)}$ be such a subsequence. Then $c_k^{(j)} = c_{k+1}^{(j)} = \cdots = c_{k+\ell-1}^{(j)}$ and either $k = 1$ or $c_{k-1}^{(j)} \neq c_k^{(j)}$ and either $k + \ell - 1 = \|\mathbf{x}\|$ or $c_{k+\ell-1}^{(j)} \neq c_{k+\ell}^{(j)}$. When we use the term *subsequence* in the following we mean such a maximal subsequence. We denote the three collections of subsequences by $\mathcal{S}_H$, $\mathcal{S}_E$, and $\mathcal{S}_C$. On each of these sets a Markov model is trained by estimating its parameters. Let $\mathcal{M}_H$, $\mathcal{M}_E$, and $\mathcal{M}_C$ be the respective models. The estimations are the relative frequencies of (pairs of) residues in the training data. To avoid zero empirical probabilities, we introduce a pseudocount value $c \geq 0$, where $c = 0$ is the estimation without pseudocounts. Let $X \in \{H, E, C\}$ be the class and let $\mathbf{y}^{(j)}$ denote the maximal subsequences. Then the estimations are

$$p_X(a) := \frac{c + \left|\{j \mid \mathbf{y}^{(j)} \in \mathcal{S}_X \wedge y_1^{(j)} = a\}\right|}{|\Sigma_a|\, c + |\mathcal{S}_X|} \tag{3}$$

$$p_X(b|a) := \frac{c + \left|\left\{(i,j) \mid \mathbf{y}^{(j)} \in \mathcal{S}_X \wedge 1 < i \leq \left\|\mathbf{y}^{(j)}\right\| \wedge y_i^{(j)} = a \wedge y_{i-1}^{(j)} = b\right\}\right|}{\left|\left\{(i,j) \mid \mathbf{y}^{(j)} \in \mathcal{S}_X \wedge 1 < i \leq \left\|\mathbf{y}^{(j)}\right\| \wedge y_i^{(j)} = a\right\}\right| + |\Sigma_a|\, c} \tag{4}$$

3 Improvements

The basic classification method described above has been analysed and modified in order to detect the importance of the various parameters and to improve its performance. The tests were carried out while maintaining the statistical nature of the approach. No biological background knowledge was incorporated. Also, the method was not combined with other techniques. The aim was to push the performance of the basic method as far as possible before applying other techniques. In the following we describe the modifications and their influence on the performance.

The results shown here come from tests performed in Larsen and Thomsen (2004) on the GOR data set (Garnier et al. (1996)), which consists of 267 protein sequences. It was evaluated using a leave-one-out cross-validation. We also used the benchmark data set of 513 protein sequences. The results on the latter set showed no relevant difference to those on the GOR data set. Due to the structure of the Markov model with window size ℓ, the last $\ell - 1$ residuals of a sequence cannot be classified. The percent figures thus are the ratios of correctly classified residuals and all classified residuals.

Pseudocount and window size: These two parameters have been varied independently. The window size parameter ℓ is the number of terms used in the Markov expansion (1). The range for the window size was 1 through 10.

One would expect that a very small window size results in bad performance, because too few information is used in the classification process. Also very large window sizes should decrease the performance because the local information is blurred by far off data. The pseudocount parameter c was varied from 0 through 1000. The effect of this parameter depends on the size of the training set. In our case the set was so large, that no zero empirical probabilities occurred. Nevertheless, the performance of the classifier was improved when using small positive pseudocount values. We believe that this is due to the fact, that statistical fluctuation in small (unprecisely estimated) empirical probabilities are leveled by this.

The optimal choice of the parameters was a window size of 5 and a pseudocount value of 5. These settings were used in all following results. We also varied the window size and pseudocount constant in combination with other modifications but the aforementioned values stayed optimal. Figure 1 shows a plot of the test results. With this choice, the basic model has a classification rate (number of correctly classified residuals) of 51.0%. The naive classification – constantly predicting the most frequent residual (coil) – would give 43%.

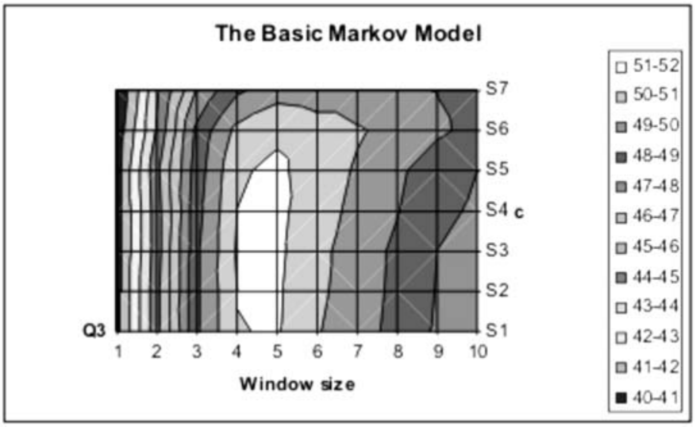

Fig. 1. A contour plot of the prediction performance of the Markov model as a function of the window size and the pseudocount constant. The vertical axis is from $S1 = 0$ to $S7 = 15$

Estimation of $p_X(a)$: In Equation (3) the parameter $p_X(a)$ was estimated as the empirical frequency of a as a first letter of a maximal subsequence with classification X. This definition stems from the application in Brunnert et al. (2002) where additional knowledge on the structure of the subsequences (length/order) was available. We changed the estimation (3) to

$$p'_X(a) := \frac{c + \left|\{(i,j) \mid \mathbf{y}^{(j)} \in \mathcal{S}_X \wedge y_i^{(j)} = a\}\right|}{|\Sigma_a|\, c + \sum_j \left|\mathbf{y}^{(j)}\right|} \ , \tag{5}$$

the frequency of the letter a in all subsequences with classification X. Using this definition improved the classification performance by 1.4 percentage points. The increase was expected, because the information on residuals in the middle of the subsequence is increased.

Estimation of $p(a|b)$: Instead of using Equation (4) to estimate the conditional probabilities, we also considered the reversed sequence. That is we computed $p^{forw}(a|b)$ as in Equation (4) and $p^{rev}(a|b)$ as in Equation (4) but on the reversed sequence. Then we set $p(a|b)$ to the sum of $p^{fore}(a|b)$ and $p^{rev}(a|b)$ and normalize to get a probability distribution. Using this definition of $p(a|b)$ improved the prediction performance by 2 percentage points.

Direction: Markov models exploit directional information. We therefore tried another modification, namely to reverse the sequences in the training and the classification process. We did not expect a significant increase from this. To our surprise the classification performance was increased by 1.5 percentage points when using reversed sequences. This indicates that the sequence data is more informative in one direction than in the other one.

Momentum: This variation of the basic method tries to achieve a more "stable" classification as the classification window moves along the DNA-sequence. To this end we consider the discounted values of previous classifications. The new classification value, denoted by $p'_X(i)$, replaces the original values $p_X(i)$ from (2) and is defined by

$$\begin{aligned} p'_X(1) &= p_X(1) \\ p'_X(i) &= w \cdot p'_X(i-1) + (1-w) \cdot p_X(i) \end{aligned}$$

To determine a good value for the discount constant w, different settings of $w \in [0,1]$ were tested. The choice of $w = 0.5$ showed the best results with an increase of 4.3 percentage points in the prediction performance.

One can say that the use of a momentum term does model some biological knowledge. It is known that helix, coil, or sheet subsequences usually consist of a number of amino acids, not just a single one. The momentum method eliminates a number of subsequences of length one from the prediction. This often replaces the old prediction by the correct one, which results in the better performance.

Combination of methods: A number of combinations of the above methods were tested. Combining the definition given in Equation (5) for $p(a)$, the momentum and the modified definition of $p(a|b)$ proved to be the most successful one. It resulted in the considerable increase of the prediction performance of 6.3 percentage points resulting in 57.3%.

Our implementations of the GOR algorithm versions I, III and IV, all without the incorporation of background knowledge and with window size 17, gave classification rates of 60.7%, 59.6%, and 63.4%. It is not surprising that the GOR-algorithms outperform the Markov approach, as it uses a statistic of all pairs in the window. It is however surprising, how close one can come to versions I and III of GOR.

4 Ongoing research

We are currently considering "peaks" of the probabilities. The idea of using the concept of a peak is motivated by the shapes of the graphs of the three probability functions $p_E(i)$, $p_H(i)$, and $p_C(i)$. Often the function p_X has a peak at the first residuum of a X-subsequence. See Figure 2 for an example. The peaks are more prominent when using the original definition (3) of the term $p(a)$ than that given in (5). A peak could be used as indicator of the start of a new subsequence. Then the corresponding classification is maintained until a peak of another probability function is found.

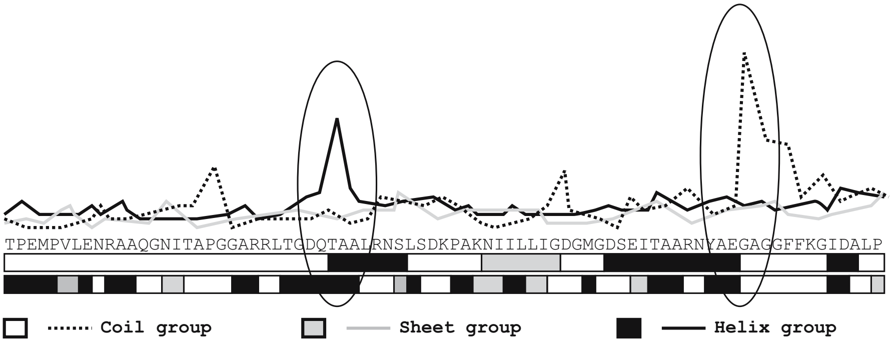

Fig. 2. The picture shows the probability functions for the three classes. Two peaks at the left start points of subsequences are marked by ovals. Below the graphs is the protein sequence with the correct classifications shown by colors. The colors in the line below show the predictions of the Markov model.

The problem here is to find an appropriate combinatorial definition of the term "peak". The absolute value of the functions p_X cannot be used due to their strong variation. Also, a peak of one function, say p_E, does not necessarily exceed the values of the two other functions. On the other hand, a peak value of p_E should not be ridiculously small relative to the two other functions.

First tests with a simple definition of a peak show that using this concept as a start indicator only gives an improvement of 4 percentage points over the naive classification leading to 47%. The plan is to incorporate peak indicators into the Markov method (or other prediction methods). One way of doing this is to compare the peak locations with a prediction given by some other method. Then the alignment of a peak with the start of a predicted subsequence would raise our confidence in the prediction. If a predicted subsequence does not coincide with a peak, then the prediction at this location should be checked.

5 Summary

We have significantly improved a simple statistical prediction method by a thorough analysis of the influence of its different components. Now, the next step is to incorporate biological background knowledge into the classification process and to combine the Markov predictor with other classifiers. The investigations also exposed the "peak" concept as a promising alternative for using the statistical information.

References

BRUNNERT, M., FISCHER, P. and URFER, W. (2002): Sequence-structure alignment using a statistical analysis of core models and dynamic programming. Technical report, SFB 475, Universität Dortmund.

GARNIER, J., GIBRAT, J.-F. and ROBSON, B. (1996): GOR method for predicting protein secondary structure from amino acid sequence. *Methods in Enzymology, 266, 540–553.*

KLOCZKOWSKI, A., TING, K.L., JERNIGAN, R.L. and GARNIER, J (2002): Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from aminoacid sequence. *Proteins, 49, 154–166.*

LARSEN, S. and THOMSEN, C. (2004): Classification of protein sequences using Markov models, Masters thesis, Informatics and Mathematical Modelling. Technical University of Denmark.

ROST, B. and SANDER, C. (1993): Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol., 232, 584–599.*

ROST, B. and SANDER, C. (1994): Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins, 19, 55–72.*