**Protein β-turn prediction using nearest-neighbor method**

Saejoon Kim

School of Computational Sciences, Korea Institute for Advanced Study, Seoul 130-722, Korea

Received on April 2, 2003; revised on May 24, 2003; accepted on July 2, 2003

**ABSTRACT**

**Motivation:** With the emerging success of protein secondary structure prediction through the applications of various statistical and machine learning techniques, similar techniques have been applied to protein β-turn prediction. In this study, we perform protein β-turn prediction using a k-nearest neighbor method, which is combined with a filter that uses predicted protein secondary structure information. Traditional β-turn prediction from k-nearest neighbor method is modified to account for the unbalanced ratio of the natural occurrence of β-turns and non-β-turns.

**Results:** Our prediction scheme is tested on a set of 426 non-homologous protein sequences. The prediction scheme consists of two stages: k-nearest neighbor method stage and filtering stage. Variations of the k-nearest neighbor method were used to take property of β-turns into consideration. Our filtering method uses β-turn/non-β-turn estimates from the k-nearest neighbor method stage and predicted protein secondary structure information from PSI-PRED in order to get new β-turn/non-β-turn estimate. Our result is compared with the previously best known β-turn prediction method on the dataset of 426 non-homologous protein sequences and is shown to give slightly superior performance at significantly lower computational complexity.

**Availability:** Contact the author for information on the source code of the programs used.

**Contact:** saejoon@kias.re.kr

**INTRODUCTION**

Protein secondary structure consists of α-helices, β-sheets and coils, where the former two are considered to be regular secondary structures and the latter one is considered to be an irregular secondary structure. Coil structure can be further decomposed into substructures that include α-turns, β-turns, δ-turns, γ-turns, π-turns, bulges and random coil structures. The turns are characterized by non-repeating backbone torsion angles and are classified according to the number of residues that form them.

β-Turn in particular plays an important role in the formation of compact globular shape of proteins. β-Turn is a four-residue reversal in a protein chain that is not within an α helix, and the distance between the first and the last (the fourth) Cα is <7 Å. They tend to be found at solvent-exposed surfaces and many of them share residues with another turn, i.e. multiple turns. About a quarter of all protein residues are in β-turns and about 58% of all β-turns are reported to be multiple turns (Hutchinson and Thornton, 1994). Since β-turns are four-residue reversals that are located at solvent-exposed surfaces, it is clear that they help in the formation of shape of proteins. Furthermore, being at solvent-exposed surfaces, the residues that form β-turns tend to be hydrophilic residues.

β-Turn forms an integral component in the fundamental building block for antiparallel β-sheets, which plays a good candidate for molecular recognition processes since being at solvent-exposed surfaces, and its formation is an important stage during the process of protein folding (Takano et al., 2000). Therefore, to improve on the identification of structural motifs such as the building block for antiparallel β-sheets and fold recognition, an accurate method to identify the location of β-turns in a protein sequence needs to be developed. Consequently, this method will serve as an important step toward predicting the overall three-dimensional structure of a protein from its one-dimensional amino acid sequence information alone.

**Previous methods**

A number of β-turn prediction methods have been developed in the past. Many of the them are empirically based and a representative of them is the positional preference approach (Lewis et al., 1973; Chou and Fasman, 1974; Wilmot and Thornton, 1988; Hutchinson and Thornton, 1994). In this approach, a set of probabilities and conformational parameters for each amino acid type occurring at each position of the β-turn is derived. This approach was extended to the 1–4 and 2–3 correlation approach (Zhang and Chou, 1997), where correlation of the pairing of the first and the fourth, and of the second and the third residues in a β-turn is taken into account. Another type of method used for β-turn prediction is the neural network approach (McGregor et al., 1989; Shepherd et al., 1999). In this approach, parameters of the neural network is learned from a training set that are used to make the prediction.
of the test set. Neural network is considered to be one of the best-performing classification methods to date.

Due to the disparity between the number of β-turn residues which comprise about 25% of all protein residues and non-β-turn residues, it is possible to get about 75% of prediction correct simply by guessing all residues to be non-β-turn. In order to circumvent this problem, a more robust performance measure called the Matthew’s correlation coefficient (MCC) is typically used along with other traditional measures. Under this MCC measure, neural network approach by McGregor et al. (1989) achieved MCC of only 0.20. To improve upon this rate, method by Shepherd et al. (1999) used predicted secondary structure information from PHDsec program to achieve MCC of about 0.35.

An evaluation of some six of the β-turn prediction methods based on a common dataset was made recently (Kaur and Raghava, 2002). Methods by Chou and Fasman (1974, 1979), Wilmut and Thornton (1988, 1990), Zhang and Chou (1997), Chou (1997), and Shepherd et al. (1999) were evaluated where all except the last method are based on positional frequencies, conformational parameters and correlations therein of the residues. The evaluation showed that neural network approach by Shepherd et al. (1999) gave the best prediction performance.

In this study, we test β-turn prediction using a k-nearest neighbor method. Nearest neighbor methods have been used in the prediction of protein secondary structures in the past (Yi and Lander, 1993; Salamov and Solovyev, 1995), and gave performance that is comparable to other protein secondary structure prediction methods. Since, most approaches for protein secondary structure prediction have been applied to β-turn prediction with the exception of the nearest neighbor method, we find it a natural next step in β-turn prediction to use a k-nearest neighbor method to test if it works comparatively well as in the case of secondary structure prediction.

SYSTEM AND METHODS
Dataset
We used the dataset of 426 non-homologous protein chains used recently by Kaur and Raghava (2002) for a performance evaluation of six β-turn prediction methods. β-Turn assignment of the residues was made by the PROMOTIF program (Hutchinson and Thornton, 1996). In this dataset, no two protein chains share >25% sequence identity, each structure of protein chain is determined by X-ray crystallography at ≥2.0 Å resolution, and each protein chain contains at least one β-turn.

Overview
β-Turn prediction was evaluated by running a partial jackknife test on 20 randomly chosen protein chains. In the full jackknife test, a protein chain is withdrawn from the whole dataset, i.e. 426 protein chains in our case, and its each residue’s β-turn/non-β-turn status is predicted using information from the remaining 425 protein chains. In other words, the β-turn status of the residues in the withdrawn protein chain is predicted using PROMOTIF program’s β-turn assignments of the residues in the remaining protein chains. The chosen protein chain is called the test set, and the set of remaining protein chains is called the training set. This process is repeated for all 426 protein chains. In our partial jack-knife test, we ran this process randomly for 20 protein chains.

We show three β-turn prediction schemes in this study, all of which consist of two stages of prediction. In our first β-turn prediction scheme, the first stage is a k-nearest neighbor method and the second stage is a filter, which refines the prediction by taking correlations amongst residues into account that k-nearest neighbor method alone does not do. These will be described in detail in ‘Algorithm’ section.

In the second β-turn prediction scheme, to observe the significance of disparity between the number of β-turns and non-β-turns to the k-nearest neighbor method, a modified k-nearest neighbor method in the first stage followed by the filter second stage is employed for the prediction. Finally, to see if different residues in a protein fragment have different importance in the k-nearest neighbor method, input data is multiplied by unequal weighing factor before applying the second β-turn prediction scheme which is our third β-turn prediction scheme. These will be described in detail in ‘Algorithm’ section.

ALGORITHM
As described above, there are three algorithms that we show in this study. The three algorithms all consist of two stages where the first stage is a type of k-nearest neighbor method, and the second stage is a filter. Following three subsections describe the first stage of the three algorithms, respectively, and the fourth subsection describes the second stage of the three algorithms which is common to all three algorithms.

k-nearest neighbor method
In this k-nearest neighbor method, each residue in a protein chain is represented by a fixed length window of residues in the protein chain where a ‘null’ residue was used to fill in the empty position. The fixed length window of residues will be called protein fragment from here on, and protein fragment from the test set and the training set will be called the test fragment and the training fragment, respectively. The amino acid type of each residue is encoded by using the unary encoding scheme where each amino acid type is translated into a length-20 vector of a single one and 19 zeros. For example, in our encoding scheme, alanine was encoded to (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)T where T stands for transpose. The ‘null’ residue was represented by the all-zero length-20 vector. Hence, a protein fragment of window size W is represented by a 20 by W matrix of zeros and ones.
The distance between two protein fragments is the usual sum of distances between entries of two fragments’ matrices. For example if the two fragments’ matrices are represented by $A = (a_{ij})$ and $B = (b_{ij})$, then the distance between the two fragments is $\sum_{i,j} |a_{ij} - b_{ij}|$.

The $k$-nearest neighbor method is the following. Given a residue and its associated test fragment, obtain the test fragment’s $k$-closest training fragments. If more than half the closest training fragments are associated with residue of $\beta$-turn assignment, residue of the test fragment is estimated to be a $\beta$-turn, and vice versa.

**Handling unbalanced data**

A property of our dataset is that the number of $\beta$-turns is comparatively small (about one in four residues) to non-$\beta$-turns. This translates into the $k$-nearest neighbor method tending to overpredict the larger class of non-$\beta$-turns. On the other hand, if a naturally occurring proportion of $\beta$-turns and non-$\beta$-turns exist in the test set, and a balanced proportion of $\beta$-turns and non-$\beta$-turns are selected to be in the training set, then the nearest neighbor method will underpredict the larger class of non-$\beta$-turns. The former case is called unbalanced training set and the latter case is called balanced training set. In order to take both unbalanced and balanced training sets into consideration, we have made a variation in the fraction of training fragments associated with a $\beta$-turn assignment required for the residue of the test fragment to be predicted as a $\beta$-turn. For the first stage of our second $\beta$-turn prediction algorithm, instead of requiring more than $k/2$ closest training fragments associated with a $\beta$-turn assignment to predict the residue of the test fragment as a $\beta$-turn, we have tested on a various number of minimally required closest training fragments.

**Unequal weighing of data**

We gave different weights in columns of the matrices of both the training and the test protein fragments to see if residues in the center of protein fragment play more important role than the ones towards the edge of the fragment in the $\beta$-turn prediction. This was motivated from our research in protein secondary structure prediction in which multiplying by unequal weights in the columns of the matrix increased a couple of percentage points in the prediction rate. For the first stage of our third $\beta$-turn prediction algorithm, the columns of protein fragments’ matrices are multiplied by weights which is increasing monotonically going from the first or the last column to the middle column, and then such matrices are used as input to the modified $k$-nearest neighbor method used in our second $\beta$-turn prediction algorithm.

**Filter**

To exploit the fact that $\beta$-turns are four residues long, we added a filtering stage in our prediction scheme. The predictions from the first stage are not correlated and they are simply based on $k$-nearest neighbor method for each residue separately without any reference to neighboring residues’ $\beta$-turn/non-$\beta$-turn status.

To take advantage of the fact that $\beta$-turns are located mostly in the coil region of a protein chain, we used the secondary structure prediction method of PSI-PRED (Jones, 1999) to identify those residues that are predicted as non-coils. Given a protein chain, its residues’ $\beta$-turn/non-$\beta$-turn predictions from the first stage, and its residues’ secondary structure predictions from PSI-PRED, our second stage filters all those residues predicted to be non-coil to be non-$\beta$-turn.

Furthermore, since $\beta$-turns are typically multiple turns of at least four residues long, the second stage adopts another set of rules similar to the state-flipping rule used in Shepherd *et al.* (1999). First, the second stage filters all isolated non-$\beta$-turn predicted residues to be $\beta$-turns. Then it filters all those residues that are either isolated $\beta$-turn predicted residues or isolated pairs of $\beta$-turn predicted residues to be non-$\beta$-turns. Finally, for the isolated three consecutive $\beta$-turn predicted residues, the two adjacent non-$\beta$-turn predicted residues are filtered to be $\beta$-turns. This ensures $\beta$-turn predictions to be at least four residues long.

**Performance measures**

There are various ways to measure the prediction performance where some are more suitable than others depending on the application considered. In accordance with a number of previous papers on $\beta$-turn prediction, we give four widely used measures here. Following is a list of four quantities in calculating the performance measures: (1) $tp$, number of correctly classified $\beta$-turn residues; (2) $tn$, number of correctly classified non-$\beta$-turn residues; (3) $fp$, number of incorrectly classified $\beta$-turn residues and (4) $fn$, number of incorrectly classified non-$\beta$-turn residues. $tp$, $tn$, $fp$ and $fn$ stand for true positive, true negative, false positive and false negative, respectively. The most common measure of overall performance is $Q_{\text{total}}$ which is the fraction of correctly predicted $\beta$-turns and non-$\beta$-turns among all predictions.

$$Q_{\text{total}} = \frac{tp + tn}{tp + tn + fp + fn} \times 100.$$  

To get a measure on the sensitivity of prediction performance, $Q_{\text{pred}}$, which is the fraction of correctly predicted $\beta$-turns among predicted $\beta$-turns, is used.

$$Q_{\text{pred}} = \frac{tp}{tp + fp} \times 100.$$  

Similarly, to get a measure on the selectivity of prediction performance, $Q_{\text{obs}}$, which is the fraction of correctly predicted $\beta$-turns among observed $\beta$-turns, is used.

$$Q_{\text{obs}} = \frac{tp}{tp + fn} \times 100.$$
To get a measure that displays both sensitivity and selectivity of prediction performance, MCC is typically used.

$$MCC = \frac{(tp \cdot tn) - (fp \cdot fn)}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}}$$

All the above measures can be meaningless if they are worse than a random prediction. The following is the normalized score better than random (S) (Shepherd et al., 1999).

$$S = \frac{(tp + tn) - R}{tp + tn + fp + fn - R},$$

where

$$R = \frac{(tp + fp)(tp + fn) + (tn + fp)(tn + fn)}{tp + tn + fp + fn}.$$

Clearly, $R$ measures the expected number of residues that would be classified correctly by a random prediction. Expression for $S$ shows that $S = 1$ for perfect prediction and $S = 0$ for prediction that is no better than a random prediction.

**RESULTS**

In this section, we give the prediction results of the three methods discussed in the ‘Algorithm’ section. In all three methods, we varied two parameters to see how they affect the prediction performance. The first parameter we consider is the number of neighbors $k$. Our data show that the distribution of test fragments do not resemble any obvious density function and, therefore, it was not easy to estimate theoretically how many neighbors are optimal for the prediction. We did not try to optimize on the number of nearest neighbors; however, listed the prediction results for two $k$’s, 10 and 50, that seemed to perform relatively well. When the number of neighbors goes greater than 100, prediction performance turns out poorly. The second parameter we varied is the window size in protein fragments. Again, we do not know theoretically the size of the local structural environment of a residue that provides the best prediction performance. We tried window sizes of 3–9 to see which performs better, where a residue was represented by two windows for even number window sizes because a residue can be located in the right or the left of the two center residues in this case. We used the notation $a$ or $b$ concatenated with the even number window size $W$ to mean window size $W$ with the residue of interest located in the $[(W/2) + 1]$th or the $(W/2)$th coordinate in the window, respectively. For window sizes greater than 11, prediction performance degraded severely.

**Prediction with predicted secondary structure information**

In Table 1, we list the performance results for the $k$-nearest neighbor method followed by a filter. Columns two through five show the performance results after the $k$-nearest neighbor method has been filtered by the secondary structure predictions from PSI-PRED and the state-flipping-like rule. The table shows the result for $k = 10$ case. We have also tried $k = 50$ case, and the results were slightly worse than $k = 10$ case, especially in the second stage. The table shows that window size of 4 performs comparatively well; however, the best result in the table is worse than that of the neural network method and slightly better than those of other previously used methods.

**Balanced prediction with predicted secondary structure information**

In Table 2, we list the performance results for a modified $k$-nearest neighbor method followed by a filter. The modification made to the $k$-nearest neighbor method is the minimal number of training fragments associated with a $\beta$-turn residue to be included in the $k$-nearest neighbors required in order for the residue of the test fragment to be predicted as a $\beta$-turn.

### Table 1. 10-nearest neighbor method with predicted secondary structure information

<table>
<thead>
<tr>
<th>Window size</th>
<th>First stage</th>
<th>Second stage</th>
<th>Second stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protein 3</td>
<td>Q_total 75.49</td>
<td>Q_total 74.72</td>
<td>Q_total 74.72</td>
</tr>
<tr>
<td>4a 55.73</td>
<td>39.53</td>
<td>39.53</td>
<td>39.53</td>
</tr>
<tr>
<td>4b 56.95</td>
<td>40.69</td>
<td>40.69</td>
<td>40.69</td>
</tr>
<tr>
<td>5 54.23</td>
<td>39.68</td>
<td>39.68</td>
<td>39.68</td>
</tr>
<tr>
<td>6a 54.14</td>
<td>39.59</td>
<td>39.59</td>
<td>39.59</td>
</tr>
<tr>
<td>6b 56.56</td>
<td>41.45</td>
<td>41.45</td>
<td>41.45</td>
</tr>
<tr>
<td>7 57.41</td>
<td>42.78</td>
<td>42.78</td>
<td>42.78</td>
</tr>
<tr>
<td>8a 57.41</td>
<td>42.78</td>
<td>42.78</td>
<td>42.78</td>
</tr>
<tr>
<td>8b 59.43</td>
<td>43.94</td>
<td>43.94</td>
<td>43.94</td>
</tr>
<tr>
<td>9 62.35</td>
<td>45.12</td>
<td>45.12</td>
<td>45.12</td>
</tr>
</tbody>
</table>

### Table 2. Modified 50-nearest neighbor method with predicted secondary structure information

<table>
<thead>
<tr>
<th>Window size</th>
<th>First stage</th>
<th>Second stage</th>
<th>Second stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protein 3</td>
<td>Q_total 75.49</td>
<td>Q_total 74.72</td>
<td>Q_total 74.72</td>
</tr>
<tr>
<td>4a 55.73</td>
<td>39.53</td>
<td>39.53</td>
<td>39.53</td>
</tr>
<tr>
<td>4b 56.95</td>
<td>40.69</td>
<td>40.69</td>
<td>40.69</td>
</tr>
<tr>
<td>5 54.23</td>
<td>39.68</td>
<td>39.68</td>
<td>39.68</td>
</tr>
<tr>
<td>6a 54.14</td>
<td>39.59</td>
<td>39.59</td>
<td>39.59</td>
</tr>
<tr>
<td>6b 56.56</td>
<td>41.45</td>
<td>41.45</td>
<td>41.45</td>
</tr>
<tr>
<td>7 57.41</td>
<td>42.78</td>
<td>42.78</td>
<td>42.78</td>
</tr>
<tr>
<td>8a 57.41</td>
<td>42.78</td>
<td>42.78</td>
<td>42.78</td>
</tr>
<tr>
<td>8b 59.43</td>
<td>43.94</td>
<td>43.94</td>
<td>43.94</td>
</tr>
<tr>
<td>9 62.35</td>
<td>45.12</td>
<td>45.12</td>
<td>45.12</td>
</tr>
</tbody>
</table>
Table 3. Performance comparison between our method and a neural network method

<table>
<thead>
<tr>
<th></th>
<th>$Q_{\text{total}}$</th>
<th>$Q_{\text{pred}}$</th>
<th>$Q_{\text{obs}}$</th>
<th>MCC</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>75.0</td>
<td>46.5</td>
<td>66.7</td>
<td>0.40</td>
<td>0.38</td>
</tr>
<tr>
<td>Neural network method</td>
<td>74.9</td>
<td>55.3</td>
<td>48.0</td>
<td>0.35</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 2 shows the result for $k = 50$ case which seemed to be slightly better than $k = 10$ case. For both modified $k$-nearest neighbor methods, $k = 10, 50$, we found that requiring about 20% of $k$-nearest neighbors (2 for $k = 10$ case and 10 for $k = 50$ case) as the minimal fraction of training fragments associated with a $\beta$-turn gave the best result. The results shown in Table 2 therefore correspond to using 10 as the minimal number of training fragments associated with a $\beta$-turn residue. Most of the results shown in Table 2 are at least as good as the best $\beta$-turn prediction results up to date. In fact, window size of 8 achieves MCC of 0.40. It is worth noting that results for two different windows of the same size are similar in this table while it differed slightly in Table 1.

**Balanced prediction of weighted data with predicted secondary structure information**

Unlike the case for secondary structure prediction, using different weights in the columns did not provide improvement to the prediction using equal weights in the columns. For example, for $k = 10$ case, use of different weights provided very small improvement, while for $k = 50$ case, it produced slightly worse results.

In Table 3, we summarize our result and compare with that of the previously best known prediction method, namely, the neural network method. We note that result of this paper is not directly comparable to that of the neural network method because the latter used slightly different dataset and secondary structure prediction scheme. It is assumed that neural network method would perform slightly better than shown in Table 3 if tested on the same dataset using the same secondary structure prediction method used in this paper. Nevertheless, the table shows that our modified $k$-nearest neighbor method followed by a filter outperforms the neural network method by a non-trivial gap, and is clearly a better prediction scheme than random guessing.

**DISCUSSION**

From a straightforward nearest neighbor method approach with $Q_{\text{total}}$, $Q_{\text{pred}}$, $Q_{\text{obs}}$ and MCC of 76.51%, 47.04%, 18.12% and 0.18, we have made improvement to the performance of 75.02%, 46.47%, 66.70% and 0.40, respectively, with our modified nearest neighbor method with filter approach. This is the currently best known $\beta$-turn prediction result based on input of amino acid sequence and its predicted secondary structure information alone.

We believe that since the approaches we took in this study are clearly distinct from other previously known $\beta$-turn prediction schemes, they can be combined to produce even better performance.

**REFERENCES**