Sparse Kernel Logistic Regression for β-turns Prediction

Murtada Khalafallah Elbashir, Jianxin Wang, Fang-Xiang Wu, Min Li.
School of Information Science and Engineering, Central South University
Changsha, 410083, P.R. China.
murtadabashir@yahoo.com, jxwang@mail.csu.edu.cn, faw341@mail.usask.ca, limin@mail.csu.edu.cn.

***Abstract*—A β-turn is a secondary protein structure type that plays a significant role in protein folding, stability, and molecular recognition. On average 25% of amino acids in protein structures are located in β-turns. Development of accurate and efficient method for β-turns prediction is very important. Most of the current successful β-turns prediction methods use support vector machines (SVMs) or Neural Networks (NNs), however a method that can yield probabilistic outcome, and has a well-defined extension to the multi-class case will be more valuable in β-turns prediction. Although kernel logistic regression (KLR) is a powerful classification technique that has been applied successfully in many classification problems, however it is often not found in β-turns classification, mainly because it is computationally expensive. In this paper we used KLR to obtain sparse β-turns prediction in short evolution time after speeding it using Nystrom approximation method. Secondary structure information and position specific scoring matrices (PSSMs) are utilized as input features. We achieved Qtotal of 80.4% and MCC of 50% on BT426 dataset. These results show that KLR method with the right algorithm can yield performance equivalent or even better than NNs and SVMs in β-turns prediction. In addition KLR yields probabilistic outcome and has a well-defined extension to multi-class case.**

***Index Terms*—beta-turn, kernel logistic regression, position specific scoring matrices, secondary structure information.**

I. Introduction

The number of known protein sequence is increasing rapidly as a result of genome and other sequencing projects. Consequently this increase widen sequence-structure gap rapidly[1], [2]. Thus computational tools for predicting protein structure and function are highly needed to narrow the widening gap[3]. there are four distinct levels of protein structures, these levels are: primary structure which refers to amino acid linear sequence of the polypeptide, secondary structure which is defined by the patterns of hydrogen bonds between backbone amide and carboxyl groups, tertiary structure which is the three dimensional structure of a single protein molecule, and quaternary structure which is a larger assembly of several protein molecules or polypeptide chains.

The basic elements of the secondary structure of proteins are α-helices, β-sheets, coils, and turns. A turn is a structural motif where the α-atoms of two residues are separated by few (usually 1 to 5) peptide bonds, and the distance between them is less than $7A^{\circ}$, while the corresponding residues do not form a regular secondary structure element such as an α-helix or β-sheet. Different turns are classified according to the separation between the two end residues. The end residues are separated by four peptide bonds in α-turns, three peptide bonds in β-turns, two peptide bonds in γ-turns, one bond in δ-turns, and five bonds in π-turns. β-turns are the most common found type of turns that constitute approximately 25% of the residue in protein. They play a significant role in protein configuration and function, and its formation is a vital stage during the protein folding. They were found to be more helpful in the context of molecular recognition and in modeling interactions between peptide substrates receptors, because they tend to be more solvent exposed than buried[4]. In the recent years it is found that β-turns are important in the design of various peptidomimetics for many diseases[5]. Therefore, development of effective, and efficient prediction methods for β-turns identification in protein will be helpful in fold recognition and drug design[6].

β-turns are further classified into different types according to the dihedral angles (φ,ψ) of the central two residues. the classification scheme proposed by Hutchinson and Thornton[26] recognizes nine distinct types of β-turn: Types I,I',II,II',VIa1,VIa2,VIb,VIII,IV. In this classification the most frequently-occurring type is type IV, which constitute approximately (35%) of the β-turns. types VIa1,VIa2, and VIb are rare types.

Most of the successful β-turns prediction methods are based on either support vector machines (SVMs) or neural networks (NNs). Ce Zheng and Lukasz[6] applied SVM based ensemble to predict β-turns, they used position specific scoring matrices (PSSMs) and secondary structure information as features in their prediction model. Petros and Jonathan [7] developed a method based on SVM, their method uses PSSMs, predicted secondary structures, and predicted dihedral angles as input features to the SVM. Adrian J et al[8] used a neural network to predict both the location and types of β-turn in protein, they incorporated secondary structure information in the features to be used as input to the NN. Kaur and Raghava[9] used two feed-forward back-propagation networks with a single hidden layer, where the first-sequence structure network is trained with the PSSMs. The initial prediction from the first network and the predicted secondary structure using PSIPRED[10], [18] are used as input to the second structure-structure network to refine the prediction obtained from the first network. Bent Petersen et al [11] presented a neural network method called NetTurnP, for predicting β-turns and β-turn types. Their method consists of two artificial neural network layers, they

used PSSMs, secondary structure, and surface accessibility as input to their model.

There is another method that can perform well as SVMs and NNs, which is the kernel logistic regression (KLR). KLR is a kernel version of logistic regression (LR). It is often not found on predicting protein secondary structures and β-turns due to its computational demand. However unlike SVMs and NNs, KLR yields a-posterior probabilities based on a maximum likelihood argument, that is beside predicting class labels, KLR provides interpretation about this labeling. When it comes to β-turn types prediction KLR has an additional advantage that its extension to multi-class classification is well described. In this paper we show that KLR can be used in predicting β-turns in an efficient and effective way using Nystrom approximation and the K-means clustering.

II. METHODS

A. Data sets

The uniform dataset of 426 nonhomologues proteins (BT426)[29], the dataset of 547 protein sequence (BT547), and the dataset of 823 protein sequence (BT823) are used to evaluate the performance of our KLR method. Several researchers used BT426 as a golden set of sequences upon which performance values are reported and compared. This dataset consists of protein chains whose structure has been determined by X-ray crystallography at a resolution of $< 2.0A^{\circ}$ or better. Each chain contains at least one β-turns region. In total 23,580 amino acids, corresponding to 24.9% of all amino acids, have been assigned to be located in β-turns. None of the sequences in the dataset shares more than 25% sequence identity. BT426 has been used by various recent β-turns prediction methods therefore; we can use it to make direct comparisons with these methods. The other two datasets BT547, and BT823 are constructed for training and testing COUDES[28].

B. Kernel logistic regression

KLR is the kernel version of logistic regression that allows non-linear probabilistic classification by constructing the logistic regression in higher dimensional feature space using kernel function $K : \chi \times \chi \rightarrow F$. The kernel function evaluates the inner product between the input vectors in the feature space, i.e. $K(x, x') = \phi(x).\phi(x')$, where $x \in \chi \in R^D$. The KLR can be constructed in the feature space such that

$$Pr(Y = -1|X = x, w) = \frac{e^{w^T\phi(x)+b}}{1 + e^{w^T\phi(x)+b}}$$

$$Pr(Y = 1|X = x, w) = p_{1i} = \frac{1}{1 + e^{w^T\phi(x)+b}} \quad (1)$$

Where w is the KLR parameters and b is the intercept term.

The panelized negative log likelihood (PNLL) is normally used to infer KLR parameters and it can be defined in the primal weight space as follows[13]:

$$min_{w,b}\frac{1}{2}w^T w + \frac{v}{2}\sum_{i=1}^{N} log(1 + exp(-y_i(w^T\phi(x_i) + b))) \quad (2)$$

Where v is the penalty term. One of the most popular techniques used to find the maximum-likelihood estimation (MLE) for the parameters of the LR model is the iteratively re-weighted least squares (IRLS) method, which use Newton-Raphson algorithm[14]. The same method can be used for KLR in the primal weight space in which the solution for w on the (c+1)th iteration using Newton-Raphson update can be given as in the following equation, given that we normally start from initial parameter w^0

$$w^{(c+1)} = w^{(c)} + s^{(c+1)} \quad (3)$$

Where $s^{(c+1)}$ in each iteration is determined by the following minimization problem

$$q^{c+1} = min_{s^{(c+1)}}\frac{1}{2}(s^{(c+1)} + w^{(c)})^T(s^{(c+1)} + w^{(c)}) + \frac{v}{2}\sum_{i=1}^{N} g_i^{(c+1)}(e_i^{(c+1)})^2,$$

such that

$$(s^{(c+1)})^T\phi(x_i) + b = z_i^{(c+1)} - e_i^{(c+1)}, i = 1, 2, ..., N \quad (4)$$

Where $z_i^{(c+1)} = \frac{(p_{y_i i}^{(c+1)} - 1)y_i}{g_i^{(c+1)}}$ and $g_i^{(c+1)} = p_{1i}^{(c+1)}(1 - p_{1i}^{(c+1)})$. The solution to equation (4) at iteration (c+1) is given by the following dual problem

$$\left(\begin{array}{c|c} \Omega + \frac{1}{v}(W^{(c+1)})^{-1} & 1_N \\ \hline 1_N^T & 0 \end{array}\right)\left(\begin{array}{c} \alpha^{(c+1)} \\ \hline b^{(c+1)} \end{array}\right) = \left(\begin{array}{c} z^{(c+1)} + \Omega\alpha^{(c)} \\ \hline 0 \end{array}\right) \quad (5)$$

where$1_N = (1, 1, ..., 1)^T, 1_N \in R^N, z^{(c+1)} = (z_1^{(c+1)}, z_2^{(c+1)}, ..., z_N^{(c+1)})^T$,
$\alpha^{(c+1)} = (\alpha_1^{(c+1)}, \alpha_2^{(c+1)}, ..., \alpha_N^{(c+1)})^T, \Omega_{ij} = K(x_i, x_j), W^{(c+1)} = diag(g_1^{(c+1)}, g_2^{(c+1)}, ..., g_N^{(c+1)})^T$

The IRLS method for large scale problem is computationally expensive, because the linear system in equation (5) must be solved for each Newton's iteration. To reduce the computation cost of IRLS we can adopt eigendecomposition of the kernel matrix K in the form.

$$K = P\Lambda P^{'} \quad (6)$$

Where $\Lambda = diag(\lambda_i)$, $\lambda_1 \geq \lambda_2 \geq ..... \geq 0$ are the eigenvalues of the matrix K, and P is the matrix of the eigenvectors that correspond to the eigenvalues. We can select the first p eigenvectors and eigenvalues from the matrices P and Λ respectively, where $p \ll N$ to approximate the eigendecomposition matrix given in equation (6). This approximation is motivated by its widely usage e.g. principal component analysis. Using this approximation the computational cost can reduced dramatically. However computing the eigendecomposition itself is also computationally expensive. Nystrom method can be used to reduce the computation cost of computing the eigendecomposition by selecting small sample of size $M \ll N$ from the training data[15] to create the eigenproblem of equation (6). Then the required eigenvectors and eigenvalues at all N points can be approximated as:

$$\widetilde{\lambda}_i^{(N)} = \frac{N}{M}\lambda_i^{(M)}, \widetilde{p}_i^{(N)} = \sqrt{\frac{M}{N}}\frac{1}{\lambda_i^{(M)}}K_{N\times M}p_i^{(l)} \quad (7)$$

978-1-4673-4398-5/12/$31.00 ©2012 IEEE

The selected $M \ll N$ from the features matrix X should minimize the mean squares error or in another words it should contain as much information as possible. Since the Nystrom low-rank approximation depends crucially on the quantization error induced by encoding the sample set with landmark points one, can simply use the clusters obtained with K-means algorithm with outliers removal as a selected vectors[16], [13]. The computation time of the KLR using Nystrom and K-mean clustering scales to $O(NM^2)$ whereas the computation time of the SVMs is $O(N^3)$.

C. features vector

The features that are used in this study include PSSMs, and secondary structure information. It has been shown that PSSMs contributed significantly to the accuracy of β-turns prediction[6], [11]. The PSSMs are in the form of 20*M, where M represents the sequence length. The PSSMs were generated using the iterative PSI-BLAST program[17] against National Center for Biotechnology Information (NCBI) non-redundant (nr) sequence database using the default parameters. The PSSMs values are scaled to values between 0 and 1. For the secondary structure information features, four secondary structure prediction methods are utilized for all protein chains. These four prediction methods are PSIPRED [18], [19], JNET [20], TRANSEC [21], and PROTEUS [21]. The secondary structures were predicted as three structures: helix, strand and coils. The predicted secondary structure information from the four secondary structure prediction methods are added to the PSSMs features. A window size of seven residues is used for the PSSMs. This is in accordance with Shepherd et al[8] who found that the optimal prediction for β-turns is achieved using window size of seven or nine. The total number of the features that are based on PSSMs and secondary structure information is (20*7+4*3=152). Similarly as in [6], another 64 features were added to the input vector, 4 of them are the confidence score of the central amino acid using the four prediction methods, 48 features representing a binary value for a specific configuration of the secondary structure using the four methods for the central and two adjacent residue, and 12 features representing the ratio between the number of residues in a given secondary structure and the window size. Thus the total number of features in the input vector is 216. Also similar to [6], features selection methods based on information gain and CHI-squared are employed to reduce the features to 90.

D. Training and testing

To test the accuracy of β-turns prediction, seven fold cross validation was performed on all the datasets. That is, these sets were randomly divided into seven subsets, each containing equal number of proteins. Each set is an unbalanced set that retains the naturally occurring proportion of β-turns and non β-turns. Five of the seven subsets were merged together to form a training set that will be used to train the KLR model. The KLR model is validated for minimum error on the sixth subset to avoid over-training. The last subset is used for testing. This process was repeated seven times to test the prediction result for each testing set. The final prediction results are taken as the average of the results from the seven testing sets.

E. Performance measures

The quality of prediction is evaluated using five measures, MCC, Qtotal, Qpredicted, Qobserved, and Specificity. These measures are consistent with the test procedures and measures applied to evaluate competing methods. Let TP (true positives) be the number of correctly classified β-turns residues, TN (true negatives) be the number of correctly classified non β-turns residues, FP(false positives) be the number of non β-turns incorrectly classified as β-turns residues and FN (false negatives) be the number of β-turns incorrectly classified as non β-turns residues. The Matthews correlation coefficient (MCC) can be calculated as [30]:

$$MCC = \frac{(TP*TN-FP*FN)}{\sqrt{(TP+FN)*(TN+FP)*(TP+FP)*(TN+FN)}} \quad (8)$$

The result of MCC is in the range of -1 and 1, where a value of 1 indicates a perfect positive correlation, a value of -1 indicates a perfect negative correlation, and a value of 0 indicates no correlation.

Qtotal (prediction accuracy), which is defined as the percentage of correctly classified residues, and it is calculated as follows:

$$Qtotal = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (9)$$

Probability of correct prediction or Qpredicted is the percentage of correctly predicted β-turns among the predicted β-turns. It is also called predicted positive value (PPV), and it is given as follows:

$$Qpredicted = \frac{TP}{TP+FP} \times 100 \quad (10)$$

Sensitivity or coverage (also known as Qobserved), is the percentage of correctly predicted β-turns among the observed β-turns or it is the fraction of the total positive samples that are correctly predicted, and it is given as follows:

$$Qobserved = \frac{TP}{TP+FN} \times 100 \quad (11)$$

Specificity is the fraction of total negative samples that are correctly predicted.

$$Specificity = \frac{TN}{TN+FP} \times 100 \quad (12)$$

To measure whether the present method performs better than random prediction, the additional measure S is calculated. This measure is the normalized percentage of correctly predicted samples better than random [8]:

$$S = \frac{(TP+TN)-R}{(TP+TN+FP+FN)-R} \times 100 \quad (13)$$

Where R is given as follows:

978-1-4673-4398-5/12/$31.00 ©2012 IEEE

$$R = \frac{(TP+FP)(TP+FN)+(TN+FP)(TN+FN)}{(TP+TN+FP+FN)} \quad (14)$$

III. Results and discussion

The selected number of vectors M where $M \ll N$ from the feature matrix affect the accuracy of the prediction. A relatively small or big M will yields low performance. To select the optimal number of vectors a 10 fold cross validation is used starting with relatively small M and adding more vectors to M until a point where adding more vectors does not improve the classification performance. Table I depicts the prediction accuracy and MCC using different values of M. The following radial basis functions (RBF) was used as kernel function.

$$K(x_i, x_i^{'}) = e^{-\gamma \left| x_i - x_i^{'} \right|^2} \quad (15)$$

Also a 10 fold cross validation is used to tune the KLR parameter v, and the kernel function parameter γ.

TABLE I
QTOTAL AND MCC FOR DIFFERNT VALUES OF SELECTED VECTORS M.

Number of selected vectors l	Qtotal	MCC
110	0.7996	0.46
120	0.7998	0.46
130	0.7997	0.46
140	0.7996	0.46
145	0.8041	0.48
150	0.8054	0.48
160	0.8057	0.48

After a short analysis of various values of threshold we set its value to 0.45 to obtain the results in table I. The Qtotal has improved slightly when the threshold value is set to 0.50, while the MCC dropped to less than 0.46. Similarly, the MCC has increased when the threshold value is set to 0.40, but at the cost of Qtotal, which will drop to less than 0.79. The number of selected vectors M in this research is set to 150 for BT426 dataset. Using this value for M we obtained a Qtotal of 0.8054, MCC of 0.48, Qpredicted of 0.59 Qobserved of 0.62, Specificity of 0.86, and S 0f 0.48. The value of S denotes that our method is much better than random prediction. The MCC is a robust and reliable performance measure that accounts for both overpredictions and underpredictions. A high MCC value indicates a high prediction performance.

To increase the performance of our KLR model further we used state changing rules. In this rules we put in our consideration that β-turns occurs in group of at least four adjacent residues. After analyzing the results obtained by the KLR prediction, the state changing rules, which will make the prediction to be more β-turn like are derived as follows:

1- Change isolated non-turn predictions to turn (i.e tnt→ ttt)

2- Change isolated turn prediction to non-turn prediction (i.e ntn→ nnn)

3- Change the residues that are neighboring two isolated turn predictions to turn (i.e nttn→ tttt)

4- if there is isolated triplet of turns predictions, then change the adjacent non-turn prediction with the highest KLR probability output to turn (i.e ntttn→ ttttn or nttt).

The above rules should be executed in orders. After applying these rules, we obtained a better performance, where the MCC has increased from 0.48 to 0.50.

TABLE II
COMPARISON OF KLR WITH OTHER RECENT β-TURNS PREDICTION METHODS ON BT426 DATASET.

Method	Qtotal	Qpred	Qobs	Specificity	MCC
KLR	80.4	58.98	65.25	85.34	0.50
BTNpred[6]	80.9	62.7	55.6	N/A	0.47
NetTurnP[11]	78.2	54.4	75.6	79.1	0.50
BetaTPred2[9]	75.5	49.8	72.3	N/A	0.43
BTPRED[8]	74.9	55.3	48.0	N/A	0.35
DEBT[7]	79.2	54.8	70.1	N/A	0.48
SVM[22]	79.8	55.6	68.9	N/A	0.47
BTSVM[23]	78.7	56.0	62.0	N/A	0.45
E-SSpred[24]	80.9	63.6	49.2	N/A	0.44
1-4 & 2-3 correlation model[25]	59.1	32.4	61.9	N/A	0.17

Table II shows the comparison between our KLR method and other best existing β-turns prediction methods. In our method, we use the same features that are used by BTNpred. Although BTNpred, and E-SSpred achieved Qtotal of 80.9, which is higher than our own, but because of the unbalanced dataset (25% β-turns) Qtotal by itself is a poor measure. In other words, one can achieve a Qtotal of 75% by predicting all the residues to be non beat-turns. Instead our method shows high MCC 0.50 compared to BTNpred 0.47 and E-SSpred 0.44. The NetturnP and our method have the highest MCC 0.50 among the other β-turns prediction methods. Other than BTNpred and E-SSpred our KLR shows the highest Qtotal.

Table III shows a comparison between our method and other β-turns prediction methods on BT547, and BT823 datasets. Our method obtained the highest MCC 0.50, 0.49 on BT547, and BT823 respectively. Our method shows stable performances on all the three datasets used.

TABLE III
COMPARISON OF KLR WITH OTHER RECENT β-TURNS PREDICTION METHODS ON BT54, AND BT823 DATASETS.

Method	Data set	Qtotal	Qpred	Qobs	MCC
KLR	BT547	80.46	59.04	65.36	0.50
BTNpred		80.5	61.6	54.2	0.45
COUDES[28]		74.6	48.7	70.4	0.42
SVM[22]		76.6	47.6	70.2	0.43
KLR	BT823	80.66	58.42	64.64	0.49
BTNpred		80.6	60.8	54.6	0.45
COUDES		74.2	47.5	69.6	0.41
SVM[22]		76.8	53.0	72.3	0.45

All of the computations for KLR were carried out using Matlab version 2010b on a computer with 3 GB RAM, and 1.86 GHz Genuine Intel dual core processor. We compared the average elapsed time of our method with the BTNpred and E-SSpred. The results of the comparison are shown in Table IV. In this comparison, we used fold 1 in all the datasets as a

978-1-4673-4398-5/12/$31.00 ©2012 IEEE

test sets and the remaining folds as a training set. Since both BTNpred and E-SSpred used SVM, we used libsvm[31] on their features. Note that both E-SSpred and BTNpred used PSSMs and secondary structure information as features. Our method used the same features that are used by BTNpred. In addition to PSSMs and secondary structure information, E-SSpred added amino acid (AA) composition generated with classical local coding scheme.

TABLE IV
COMPARISON OF THE ELLAPSED TIME IN SECONDS BETWEEN KLR, BTNPRED, AND E-SSPRED.

Data Set	KLR	BTNpred	E-sspred
BT426	753.55	11077.185	13036.415
BT547	940.55	13261.755	15726.2
BT823	683.44	18183.256	24140.072

Compared to E-SSpred and BTNpred as shown in Table IV, our method is faster by more than a factor of 14. Although the training data in BT823 is more than the training data in BT547, but its computation time using KLR is less than the computation time of BT547, that is because the number of selected vectors M for BT823 is 90, which is by far less than the number of selected vectors for BT547, which is 140. This indicates that, for a very large dataset, a very small number of selected vectors M can be sufficient to approximate its Kernel matrix, which reflects the capability of our proposed KLR model to handle large scale datasets.

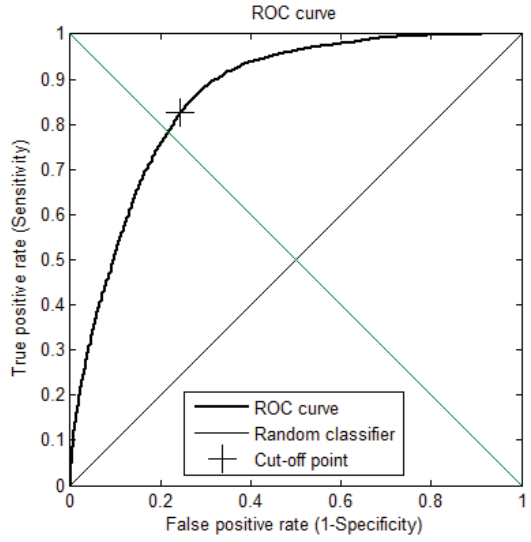

Fig. 1. ROC curve for the evaluation of the KLR model on the BT426 dataset.

The ROC curve, which is a plot of the sensitivity against the false positive rate for the evaluation of the KLR is shown in Figure 1. From the ROC curve, we calculated the area under the curve (AUC), which is a threshold independent measure. An AUC value above 0.7 is an indication of a useful prediction and a good prediction method achieves a value above 0.85[27]. NetTurnP, DEBT, E-SSpred, SVM achieved AUC of 0.864, 0.84, 0.84, 0.87 respectively; our method achieves AUC of 0.861.

IV. CONCLUSION

In this paper, we presented sparse KLR method for β-turns prediction. The Nystrom method is used to approximate the eigenvalues and eigenvectors of the Kernel matrix by selecting M vectors from the features matrix using K-means clustering algorithm. Then the first $p \ll N$ eigenvalues and eigenvectors are used to approximate the eigendecomposition matrix. Our method uses secondary structure information and PSSMs as input features. We achieved Qtotal and MCC of 80.4 and 0.50, respectively using BT426 dataset. These results are comparable and even better than the results obtained by SVMs and NNs methods. In addition, our method yields probabilistic outputs and its extension to the multi-class case is well-defined, which will be appropriate for β-turn types prediction. The computational complexity of our method is $O(NM^2)$ and its computation time is by far less than that of SVMs methods.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China under Grant No.61003124, No.61073036, No.60970095, the Ph.D. Programs Foundation of Ministry of Education of China No.20090162120073, the Freedom Explore Program of Central South University No.201012200124, and the U.S. National Science Foundation under Grants CCF-0514750, CCF-0646102, and CNS-0831634.

REFERENCES

[1] Bairoch A., Apweiler R, "The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000," Nucleic Acids Res, 28, pp. 45-48, 2000.
[2] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland1, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov and Philip E. Bourne, "The Protein Data Bank," Nucleic Acids Res, 28, pp. 235–242, 2000.
[3] Jian Guo, Hu Chen, Zhirong Sun, and Yuanlie Li, "A Novel Method for Protein Secondary Structure Prediction Using Dual-Layer SVM and Profiles," PROTEINS: Structure, Function, and Bioinformatics, 54, pp. 738–743, 2004.
[4] Rose GD, Gierasch LM, Smith JA, "Turns in peptides and proteins," Adv Protein Chem,37, pp. 100-109, 1985.
[5] Kee KS, Jois SD, "Design of beta-turn based therapeutic agents," Curr Pharm Des, 9(15), pp. 1209-1224, 2003.
[6] Ce Zheng, Lukasz Kurgan, "Prediction of beta-turns at over 80% accuracy based on an ensemble of predicted secondary structures and multiple alignments," BMC Bioinformatics 9:340, 2008.
[7] Kountouris P, Hirst J, "Predicting beta-turns and their types using predicted backbone dihedral angles and secondary structures," BMC Bioinformatics 11: 407, 2010.
[8] Shepherd AJ, Gorse D, Thornton JM, "Prediction of the location and type of beta-turns in proteins using neural networks," Protein Sci, 8, pp. 1045–1055, 1999.
[9] Kaur H, Raghava GPS, "Prediction of beta-turns in proteins from multiple alignment using neural network," Protein Sci, 12, pp. 627-634,2002.
[10] Jones, D.T, "Protein secondary structure prediction based on position specific scoring matrices," J. Mol. Biol., 292, pp. 195-202, 1999.
[11] Bent Petersen, Claus Lundegaard, Thomas Nordahl Petersen, "NetTurnP Neural Network Prediction of Beta-turns by Use of Evolutionary Information and Predicted Protein Sequence Features," PLoS ONE 5(11): e15079. doi:10.1371, 2010.

978-1-4673-4398-5/12/$31.00 ©2012 IEEE

[12] Hosmer, D.W., Lemeshow, S, Applied Logistic Regression, Wiley, 2000.
[13] P. Karsmakers "Sparse kernel-based models for speech recognition,"' PhD thesis, Katholieke Universiteit Leuven,Arenberg Doctoral School of Science, Engineering & Technology, Belgium, may 2010.
[14] Maher Maalouf, T. B. Trafalis, "Robust weighted kernel logistic regression in imbalanced and rare events data,"' Computational statistics and data analysis, 55, pp. 168–183, 2011.
[15] C.K.I. Williams, M. Seeger, "Using the Nystrom Method to Speed Up Kernel Machines,"Advances in Neural Information Processing Systems,", 13, pp. 682-688, 2001.
[16] K. Zhang, I.W. Tsang, J.T. Kwok, "Improved Nystrom low-rank approximation and error analysis," In Proc. of the 25th International Conference on Machine Learning, Helsinki, Finland, pp. 1232-1239, 2008.
[17] Altschul SF, Madden TL, Scha ffer AA, Zhang J, Zhang Z, et al, "GappedBLAST and PSI-BLAST: a new generation of protein database search programs," Nucleic Acids Res, 25, pp. 3389–3402, 1997.
[18] McGuffin LJ, Bryson K, Jones DT, "The PSIPRED protein structure prediction server," Bioinformatics, 16(4), pp. 404–405, 2000.
[19] Bryson K, McGuffin LJ, Marsden RL, Ward JJ, Sodhi JS, Jones DT, "Protein structure prediction servers at University College London," Nucl Acids Res, Web Server issue W36-38, 2005.
[20] Cuff JA, Barton GJ, "Application of multiple sequence alignment profiles to improve protein secondary structure prediction," Proteins, 40(3), pp. 502–511, 2000.
[21] Montgomerie S, Sundararaj S, Gallin WJ, Wishart DS, "Improving the accuracy of protein secondary structure prediction using structural alignment," BMC Bioinformatics 14:301, 2006.
[22] Hu X, Li Q, "Using support vector machine to predict beta- and gamma turns in proteins," J Comput Chem, 29(12), pp. 1867-1875, 2008.
[23] Pham TH, Satou K, Ho TB, "Prediction and analysis of beta-turns in proteins by support vector machine," Genome Informatics, 14, pp. 196-205, 2003.
[24] Liu L, Fang Y, Li M, Wang C, "Prediction of beta-turn in protein using ESSpred and support vector machine," Protein J., 28, pp. 175-181, 2009.
[25] Zhang C-T, Chou K-C, "Prediction of beta-turns in proteins by 1-4 and 2-3 correlation model," Biopolymers, 41, pp. 673-702, 1997.
[26] Hutchinson EG, Thornton JM., "A revised set of potentials for b-turn formation in proteins," Protein Sci, 3, pp.2207-2216, 1994.
[27] Lund O, Nielsen M, Lundegaard C, KeSmir C, Brunak S, Immunological Bioinformatics, The MIT Press: Cambridge, Massachusetts, London, England, 2005.
[28] Fuchs PF, Alix AJ, "High accuracy prediction of Beta-turns and their types using propensities and multiple alignments," Proteins, 59(4), pp. 828–839, 2005.
[29] Guruprasad K, Rajkumar S., "Beta-and gamma-turns in proteins revisited:a new set of amino acid turn-type dependent positional preferences and potentials," J Biosci, 25(2), pp.143-156, 2000.
[30] Brunak,S. ,Chauvin, Y., Andersen, C. A. F., Nielsen, H., "Assessing the accuracy of prediction algorithms: an overview," Bioinformatics, 16(5), pp. 412–424, 2000.
[31] CC C, CJ L, LIBSVM: A library for support vector machines, http: //www.csie.ntu.edu.tw/~cjlin/libsvm

978-1-4673-4398-5/12/$31.00 ©2012 IEEE