Intelligent Consensus Modeling for Proline Cis-Trans Isomerization Prediction

Paul D. Yoo, Sami Muhaidat, Kamal Taha, Jamal Bentahar, and Abdallah Shami

Abstract—Proline cis-trans isomerization (CTI) plays a key role in the rate-determining steps of protein folding. Accurate prediction of proline CTI is of great importance for the understanding of protein folding, splicing, cell signaling, and transmembrane active transport in both the human body and animals. Our goal is to develop a state-of-the-art proline CTI predictor based on a biophysically motivated intelligent consensus modeling through the use of sequence information only (i.e., position specific scores generated by PSI-BLAST). The current computational proline CTI predictors reach about 70-73 percent Q2 accuracies and about 0.40 Matthew correlation coefficient (Mcc) through the use of sequence-based evolutionary structure information as well as predicted protein secondary structure information. However, our approach that utilizes a novel decision tree-based consensus model with a powerful randomized-metalearning technique has achieved 86.58 percent Q2 accuracy and 0.74 Mcc, on the same proline CTI data set, which is a better result than those of any existing computational proline CTI predictors reported in the literature.

Index Terms—Proline cis-trans isomerization, machine-learning, intelligent systems, ensemble methods

1 INTRODUCTION

It remains an importance and relevant problem to accurately predict proline cis-trans isomers of proteins based on their amino acid sequences only. The importance of the cis-trans isomerization (CTI) as rate-determining steps in protein folding reactions has been well reported in the literature [1], [2], [3]. Prolyl CTI can be catalyzed by prolyl isomerizes, an enzyme found in both prokaryotes and eukaryotes that interconverts the cis-trans isomers of peptide bonds with the amino acid proline [4]. These enzymes are involved not only in the catalysis of folding [4], [5] but also in regulatory process [6], [7]. The CTI of prolyl peptide bonds has been suggested to dominate the folding of the alpha subunit of tryptophan synthase from Escherichia coli (aTS) [8]. A CTI, which is necessary to achieve the final conformational state of the prolyl bonds for such proteins, has often been found as the rate-limiting step in vitro protein folding [9].

The international research effort called Human Genome Project started in 1990s has produced a massive amount of biological data, and consequently, accurate and efficient computational modeling methods that can find useful patterns from the massive data have gained much attention. The first attempt to predict the CTI of proline using a computational model from amino acid sequences was made by Frömmel and Preissner in [10]. They had taken adjacent/local residues (±6) of prolyl residues and their physicochemical properties into account, and found six different patterns that allow one to assign correctly about 72.7 percent (176 cis-prolyl residues in their relatively small data set of 242 Xaa-Pro bonds of known cis-prolyl residues), where by no false positive one is predicted.

Since Frömmel and Preissner’s seminal work, support vector machines (SVMs) seemed to be the most suitable for proline CTI prediction task. The first SVM-based computational predictor was built by Wang et al. [11]. They constructed a SVM with polynomial kernel function and used amino acid sequences as input, and achieved the Q2 accuracy of 76.6 percent. Song et al. [12] also built a SVM with radial basis function, and used evolutionary information represented in position-specific-scoring matrix (PSSM) scores generated by PSI-BLAST [13] and predicted secondary structure information obtained from PSI-PRED [14] as input. They reached the Q2 accuracy of 71.5 percent, and Mcc of 0.40. Pahlke et al.’s [15] showed the importance of protein secondary structure information in the prediction of proline CTI residues. Their computational algorithm called COPS—the first attempt to predict for all 20 naturally occurring amino acids whether the peptide bond is a protein is in cis or trans conformation—used secondary structure information of amino acid triplets only. Most recently, Exarchos et al. [16] used a SVM with a wrapper feature selection algorithm, on evolutionary information (i.e., PSSM scores), predicted secondary structure information, real-valued solvent, and accessibility level for each amino acid, and the physicochemical properties of the neighboring residues as input. They achieved 70 percent accuracy in the prediction of the peptide bond conformation between any two amino acids only.

As seen above, the recent development of computational modeling for proline CTI prediction has mostly been based on SVM and its variants, and evolutionary (i.e., PSSM scores), and secondary structure information as input.
These models showed about 70-73 percent Q2 accuracies and 0.40 Mcc. This observation is aligned with the results of other computational biology studies [17], [18], [19], [20], [21]. SVMs showed great results in the prediction/classification tasks in the fields of computational biology and bioinformatics [17], [18], [19], [20], [21].

In this paper, we introduce a novel approach that utilizes biophysically-motivated intelligent consensus model (Method I) with a powerful randomized metalearning technique (Method II) through the use of sequence information only (i.e., PSSMs generated by PSI-BLAST) for the accurate and efficient prediction of proline CTI residues. The proposed model has been built based on the idea of RandomForest data modeling [22], and evolutionary information, and its predictive performance is compared with the most widely used SVM and its variants on the same data set as used in Song et al.’s study.

2 METHODS

Our experiment consists of four consecutive phases. First, collect and pre-process the proline CTI data. Second, construct each model and tune its parameters. In this phase, a standard SVM (a.k.a. Lib-SVM), a SVM variant, the proposed consensus models with Method I and II are constructed through a set of experiments that help to choose a proper kernel function and other parameters. Third, the predictive performance of the proposed methods is compared with those of SVM_LIB (Lib-SVM) and SVM_ADA (Adaboosted-Lib-SVM) for Q2 accuracy, Sensitivity (Sn), Specificity (Sp), Mathew’s correlation coefficient (Mcc), Type I and II Error Rates, and StDev for Model stability/generalization ability on the proline CTI data set built in the first phase. Lastly, we then compare those results with the consensus results from literature.

2.1 Evolutionary Data Set Construction

To make a fair comparison with existing proline CTI prediction models, we have chosen Song et al.’s [12] data set. The data set has 2,424 non-homologous protein chains, obtained from the Cucled PDB list provided by PSICES server [23]. All the tertiary structures in the data set were determined by X-ray crystallography method with resolution better than 2.0 Å and R-factor less than 0.25. In addition, the sequence identity of each pair of sequences is less than 25 percent, and the protein chains with sequence length shorter than 60 amino acids were excluded in the data set. In total, there are 609,182 residues, and every sequence contains at least one proline residue. The PDB codes, CisPep PDB codes, proline cis peptide records, corresponding dihedral angles and protein sequences of the 2,424 protein chains used in this study are available on request.

In addition, evolutionary information in the form of PSSMs was included in the windows as direct input. Evolutionary information in form of PSSMs is the most widely used input form for protein structure prediction in 1-D, 2-D and 3-D, as well as other computational/structural proteomic prediction/classification tasks [14], [15], [16], [17], [18], [19], [20], [21]. The idea of using evolutionary information in the form of PSSMs was first proposed by Jones [24], and it has improved its prediction accuracy about 3-5 percent in their prediction tasks.

To generate PSSM scores, we used the nr (non-redundant) database and blastp program obtained from NCBI [25]. We run blastp program to query each protein in our data set against the nr database to generate the PSSMs with the following setup: 1) three iterations, 2) cutoff e-value of 0.001. Finally, the PSSM scores were scaled to the range between 0 and 1 by the following standard logistic function:

\[ f(x) = \frac{1}{1 + \exp(-x)} \]

where \( x \) is the raw profile matrix value. The scaled PSSM scores were used as direct input to the learning models. A PSSM is generated for each protein sequence, and has a \( M \times 20 \) matrix, where \( M \) is the target sequence length, and 20 is the number of amino acid types. Each element of the matrix represents the log-odds score of each amino acid at one position in the multiple alignments. The window size \( 2l + 1 \) indicates the scope of the vicinity of the target prolyl peptide bonds, determining how much neighboring sequence information is included in the prediction. We selected the windows size \( (l) \) of 9, and built our models as it produced the best predictive results, aligned with Song et al.’s experimental result.

When a large difference between positive and negative samples is observed in training set, data imbalance problem exists [26]. Our data set is composed of 1,265 cis and 27,196 trans residues. There are two general approaches to reduce such imbalance problem. First, increasing the number of under-samples by random resampling. Second, decreasing the number of over-samples by random removal. In this study, we adopted the first approach, and made 1 to 1 ratios between the sizes of positive (cis) and negative (trans) training samples.

2.2 Protein Secondary Structure Information

The recent computational proteomic studies report that protein secondary structure information is useful in various protein sequence-based classifications/predictions [14], [15], [16], [17], [18], [19], [20], [21]. Although the mutations at sequence level can obscure the similarity between homologs, the secondary-structure patterns of the sequence remain conserved. That is because changes at the structural level are less tolerated. The recent studies mostly use the probability matrix of secondary structure states predicted from PSI-PRED [14]. PSI-PRED is a well-known computational predictor, and it predicts protein secondary structures in three different states (α-helix, β-sheet, and loop). However, there is one significant limitation with using predicted secondary structure information. The best secondary-structure prediction model still cannot reach the upper boundary of its prediction accuracy. In other words, it is not good enough yet to be used as a confirmation tool. It shows about 75-80 percent Q3 accuracies only. Clearly, incorrectly predicted secondary structure information if presented in input data set of a computational prediction/classification model leads to the poor learning and, eventually to the incorrect prediction of proline CTI residues. Although predicted secondary information may be useful in some extent, it should not be used if one attempts to reach...
better than 80 percent Q2 accuracy. We therefore, used evolutionary information in the form of PSSMs obtained from protein amino-acid sequences only. To achieve above 80 percent Q2 accuracy, we believe that accurate and correct information encoding presented in input data set is critical, especially if used with intelligent/model-free modeling like machine-learning. In other words, noise presented in input data set could lead to significant degrading in the performance of the models.

2.3 Method I: Intelligent Voting

This new intelligent voting/consensus approach to RandomForest data modeling combines a number of methods and procedures to exploit homology information effectively. If we take a large collection of weak learners, each performing only better than chance, then by putting them together, it is possible to make an ensemble learner that can perform arbitrarily well. Randomness is introduced by bootstrap resampling [27] to grow each tree in the ensemble learner, and also by finding the best splitter at each node within a randomly selected subset of inputs. Method I grows many decision trees (DTs) [28] as in Fig. 1. To classify a new input vector \( x \), put the input vector down each of the DTs in the ensemble learner. Each DT is trained on a bootstrap sample of the training data.

To estimate the performance of the ensemble learner, Method I performs a kind of cross-validation by using Out-of-Bag (OOB) data. Since each DT in the ensemble grows on a bootstrap sample of the data, the sequences left out of the bootstrap sample, the OOB data, can be used as legitimate test set for that tree. On average \( 1 - e^{-1} \approx 1/3 \) of the training data will be OOB for a given tree. Consequently, each PSSM in the training data set will be left out of 1/3 of the trees in the ensemble, and use these OOB predictions to estimate the error rate of the full ensemble.

Like CART [29], Method I uses the gini index for determining the final class in each DT. The gini index of node impurity is the measure most commonly chosen for classification-type problems. If a data set \( T \) contains examples from \( n \) classes, gini index \( G(T) \) is defined as

\[
G(T) = 1 - \sum_{j=1}^{n} (P_j)^2,
\]

where \( p_j \) is the relative frequency of class \( j \) in \( T \). If a data set \( T \) is split into two subsets \( T_1 \) and \( T_2 \) with sizes \( N_1 \) and \( N_2 \) respectively, the gini index of the split data contains examples from \( n \) classes, the \( G(T) \) is defined as

\[
G_{\text{split}}(T) = \frac{N_1}{N} G(T_1) + \frac{N_2}{N} G(T_2).
\]

The attribute value that provides the smallest \( G_{\text{split}}(T) \) is chosen to split the node.

Fig. 2 shows the key three steps of the Method I Ensemble. First, a random seed is chosen which pulls out at random a collection of samples from the training data set while maintaining the class distribution. Second, with this selected data set, a random set of attributes from the original data set is chosen based on user defined values. All the input variables are not considered because of enormous computation and high chances of overfitting. In a data set where \( M \) is the total number of input attributes in the data set, only \( R \) attributes are chosen at random for each tree where \( R < M \).

Third, the attributes from this set create the best possible split using the gini index to develop a DT model. The process repeats for each of the branches until the termination condition stating that leaves are the nodes that are too small to split. In this study, Method I Ensemble was constructed and implemented with the Weka RF package [30].

2.4 Method II: Randomized Metalearning

Method II builds an ensemble of randomized base classifiers (i.e., Method I), and averages their classification. Each one is based on the same input data, but uses a different random-number seed. Some learning algorithms already have a built-in random component. For example, when learning multiplayer perceptrons using the backpropagation algorithm, the initial network weights are set
to small randomly chosen values. The learned classifier depends on the random numbers because the algorithm may find a different local minimum of the error function. One way to make the outcome of classification more stable is to run the learner several times with different random number seeds (i.e., initial weights) and combine the classifiers’ predictions by voting or averaging. Learning in Method I builds a randomized DT in each iteration of the bagging algorithm, and often produces excellent predictors. Although bagging [27] and randomization yield similar results, it sometimes pays to combine them because they introduce randomness in different, perhaps complementary, ways. Randomization demands more work than bagging because the learning algorithm must be modified, but it can profitably be applied to a greater variety of learners.

The Method II input vector is formed by three seeds, and ten number of iterations. The steps involved in decision-making are depicted in Fig. 2. Tables 1 and 2 show their experimental results related to Methods I and II predictions. Using Method II to combine all the DTs, the system reduced Type I error rate significantly as depicted in Table 2.

### 2.5 Model Validation and Testing

For the system model to be useful, it must be validated to ensure that it emulates the actual system in the desired manner. This is especially true for empirical models, such as statistical machine-learning models, which primarily rely on observed data rather than analytical equations derived from first principles. The validation of these models using problem-specific information, such as theoretical relationships or experimental knowledge, should be performed. There are several methods to perform the validation task. The most common statistical methods are re-substitution, cross-validation, bootstrapping, and their variants.

To accurately assess the predictive performance of each model, we adopted a cross-validation scheme for our model evaluation. First, we apply the holdout method to our proline CTI data set. However, the holdout method has a key drawback in that the single random division of a sample into training and testing sets may introduce bias in model selection and evaluation. Since the estimated classification rate can be very different depending on the characteristic of the data, the holdout estimate can be misleading if we happen to get an unfortunate split. Hence, in our experiment, we adopted multiple train-and-test experiments to overcome the limitation of the holdout method. We created 7 to 11-fold data set, and only one of each fold was used for testing. The result of each fold is provided in Tables 1 and 2.

### 2.6 Parameter Tuning

All of the stages contain parameters or variables that need to be given appropriate values. Some of these parameters are so delicate that they have to be selected by an expert in the field, and kept constant thereafter. However, profoundly more interesting are the parameters the system is able to learn autonomously from training with available data. In our experiments, we used a semi-autonomous approach. We first used the Weka’s meta-learner, CVParameterSelection searches, and again checked the neighboring values of the best parameters found by the search. The list of the full parameters that we have used in our experiments is provided with Table 1.
3 Model Evaluation and Analysis

The performance of the models used in this study are measured by the accuracy (Q2: the proportion of true-positive and true-negative residues with respect to the total positives and negatives residues), the sensitivity (Sn: also called recall, the proportion of correctly predicted isomerization residues with respect to the total positively identified residues), the specificity (Sp: also called precision, the proportion of incorrectly predicted isomerization residues with respect to the total number of proline isomerization residues), and Mathew's correlation coefficient (Mcc: a correlation coefficient between the observed and predicted binary classifications, between -1 and +1). In Mcc, a coefficient of +1 represents a perfect prediction, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation. Hence, a high value of Mcc means that the model is regarded as a more robust prediction model. The above measures can be obtained using the following.

\[
Q_2 = \frac{TP + TN}{TP + TN + FP + FN},
\]

\[
Sp = \frac{TN}{TN + FP},
\]

\[
Sn = \frac{TP}{TP + FN},
\]

\[
Mcc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}},
\]

where TP is the number of true positives, FN is the number of false negatives or under-predictions, TN is the number of true negatives, and FP is the number of false positives or over-predictions. We adopted the polynomial kernel function and radial basis function (rbf kernel) to construct the SVM classifiers, which is aligned with the existing proline CTI prediction studies [12].

\[
K(\bar{x}_i, \bar{x}_j + 1)^d,
\]

\[
K(\bar{x}_i, \bar{x}_j) = \exp(-r\|\bar{x}_i - \bar{x}_j\|^2),
\]

where the degree \(d\) needs to be tuned as for polynomial function, and the gamma and the regulator parameters for RBF need to be regulated. See the footnote of Table 1 for the parameters settings used for this study. For the optimal learning of the prediction models, the most suitable data fold for each model should be sought.

Table 1 shows the comparison of our proposed methods results with those of SVMLEB and its variant, SVMAB. The best score in each category is underlined, and the best fold scores in each model are bolded. As seen from the table, SVM models and Method II perform better on nine-fold while Method I performs better on 10-fold. The proposed methods (Methods I and II) performed a far better then SVM models. Method I and II achieved 81.5 percent, and 86.58 percent Q2 accuracies respectively, while SVM models achieved about 76 percent Q2 accuracy only.

Table 1 also shows that our proposed methods are superior to SVM models in terms of Sp and Mcc, which indicate the model robustness and stableness. The Q2 accuracy of 86.58 percent that we achieved on proline CTI prediction is a far better than those of any existing computational proline CTI predictors reported in the literature. The best Q2 accuracy that we have found in the literature was about 73 percent on the same data set as used in this research.

The performance of each model is measured by Types I and II Error rates as well, since incorrectly predicted residues can be as valuable, as are the correctly predicted residues for further modification of the model. Type I Errors mean experimentally verified trans residues that are predicted (incorrectly) to be cis residues; type II errors indicate experimentally verified cis residues that are predicted (incorrectly) to be trans residues. Method II shows the lowest Type I Error Rate (0.21833) while SVMAB reaches the lowest Type II Error Rate (0.035). Although our proposed methods seem to be not very useful in improving Type II Error Rate, it reduces Type I Error Rate effectively. Interestingly, Type I Error Rates are worse with SVMs while Type II Error Rates are worse with our proposed methods.

StDev provides a good idea on model generalization ability. Although nonparametric machine-learning models have been proved to be useful in many different applications, their generalization capacity has often been shown to be unreliable because of the potential for overfitting. The symptom of overfitting is that the model fits the training sample too well, and thus the model output becomes unstable for prediction. On the other hand, a more stable model, such as a linear model, may not learn enough about the underlying relationship, resulting in underfitting the data. It is clear that both underfitting and overfitting will affect the generalization capacity of a model. The underfitting and overfitting problems in many data-modeling procedures can be analyzed through the well-known bias-plus-variance decomposition of the prediction error. The best generalization ability came from Method I, where a multiple DTs have been built.

The idea used in Method II using a different random-number seed seems to be useful in better learning; however, not enough to improve its generalization ability. Method I shows the best StDev value of 0.0182, while other models reach about 0.022-0.025. Fig. 3a depicts the performance comparisons of four different models in Q2, Sp, Sn, and
Mcc. As you can see, Method II outperforms other models in Q2, Sp, and Mcc, and no significant differences observed in Sn, while Method I and II improve Sp and Mcc significantly. As in Fig. 3b, Type II Error Rates are much lower than Type I Error Rates in general, and Method I and II effectively reduce Type I Error Rate. Fig. 1c shows that Method II improves in Q2, Sp, and Mcc and Type I Error Rate. However, no significant improvement observed in Sn and Type II Error Rate and StDev. This result indicates that our proposed methods could be improved by reducing the errors of experimentally verified cis residues that are predicted (incorrectly) to be trans residues.

Fig. 3d compares the model generalization ability and stability. Method I clearly outperforms other models. Again, using a different random-number seed does not really make the outcome of classification more stable, which contradicts to the findings in [31].

Although our proposed methods have shown to be useful for proline CTI prediction tasks, we suggest the following to be taken into account for the sake of improvement. First, since our methods use PSSMs only as input, homology information presented in PSSMs may not have enough information to reach the upper-boundary accuracy. Recent studies suggest that global sequence homology is seen as a strong indicator for the occurrence of prolyl cis residues [32], meaning that accurate descriptors of CTI residues and their corresponding encoding schemes must be identified. Second, solvent accessibility as a new possible input feature of proline CTI must be well examined as proline cis residues are more frequently found in surface accessible areas compared to trans residues [32]. Computational machine-learning approaches build their models based on input data only. Clearly, missing useful information in input data set leads to misclassification.

4 Conclusion

In this paper, we have presented a novel ensemble method to predict proline CTI residues in proteins. The proposed models are trained using a RF-like ensemble method, which grows a multiple trees, and chooses the classification having the most votes over all the trees in the forest, and build an ensemble of randomized base classifiers using using a different random-number seed, and averages their classification. On average, our methods are able to predict proline CTI with the Q2 accuracy of 86.58 percent, a far better than any existing proline CTI predictors reported in the literature. Experimental results on proline CTI prediction could be subjective as other existing prediction methods usually do predictions on their own data set. However, our experiments were able to demonstrate that the proposed methods can achieve a test error better than the most widely used computational models, SVM and its variants, in the literature on the same data set used in [12]. It has also demonstrated that pure evolutionary information in the format of PSSM scores as input works greatly in reducing error rate during the model learning process, meaning that noise presented (i.e., predicted secondary information) in input data set may lead to significant degrading in the performance of the models.

Acknowledgments

The authors are grateful to Dr. Song at the ARC Centre of Excellence in Structural and Functional Microbial Genomics at Monash University, Australia, for providing us the proline cis-trans isomerization data set, as well as her invaluable discussions.

References

Paul D. Yoo received the PhD degree in engineering and information technology from the University of Sydney (USyd) in 2008. He was a research fellow in the Centre for Distributed and High Performance Computing, at USyd from 2008 to 2009, and PhD researcher (Quantitative Analysis) at the Capital Markets CRC, administered by the Australia Federal Department, for Education, Science and Training, from 2004 to 2008. He is currently an assistant professor of intelligent systems and data mining in ECE Department, Khalifa University, Abu Dhabi. His recent research is centered on the application of various computer science and mathematical methods to the discovery of the syntactic and semantic patterns in nucleic acid and amino acid sequences. He holds more than 30 prestigious journal and conference publications and is currently actively involved in technical program committees, and review panels of the intelligent systems/machine-learning areas for international conference and journal publications such as IEEE, ACM, and ISCB.

Sami Muhaidat received the PhD degree in electrical and computer engineering from the University of Waterloo, Ontario, in 2006. From 2007 to 2008, he was an NSERC postdoctoral fellow in the Department of Electrical and Computer Engineering, University of Toronto, Canada. From 2008 to 2012, he was an assistant professor in the School of Engineering Science, Simon Fraser University, British Columbia, Canada. He is currently an assistant professor at Khalifa University and a visiting reader in the Faculty of Engineering, University of Surrey, United Kingdom. His research interest include advanced digital signal processing techniques for image processing and communications, machine learning, cooperative communications, vehicular communications, MIMO, and space-time coding. He has authored more than 100 journal and conference papers on these topics. He currently serves as an editor for IEEE Communications Letters and an associate editor for IEEE Transactions on Vehicular Technology. He received several scholarships during his undergraduate and graduate studies. He was also a winner of the 2006 NSERC Postdoctoral Fellowship competition. He is a senior member of the IEEE.