Published in final edited form as:
*Proteomics*. 2011 October ; 11(19): 3786–3792. doi:10.1002/pmic.201100196.

# Protein 8-class secondary structure prediction using conditional neural fields

**Zhiyong Wang**, **Feng Zhao**, **Jian Peng**, and **Jinbo Xu**
Toyota Technological Institute at Chicago, Chicago, IL, USA

## Abstract

Compared with the protein 3-class secondary structure (SS) prediction, the 8-class prediction gains less attention and is also much more challenging, especially for proteins with few sequence homologs. This paper presents a new probabilistic method for 8-class SS prediction using conditional neural fields (CNFs), a recently invented probabilistic graphical model. This CNF method not only models the complex relationship between sequence features and SS, but also exploits the interdependency among SS types of adjacent residues. In addition to sequence profiles, our method also makes use of non-evolutionary information for SS prediction. Tested on the CB513 and RS126 data sets, our method achieves Q8 accuracy of 64.9 and 64.7%, respectively, which are much better than the SSpro8 web server (51.0 and 48.0%, respectively). Our method can also be used to predict other structure properties (e.g. solvent accessibility) of a protein or the SS of RNA.

### Keywords

Bioinformatics; Conditional neural fields; Eight class; Protein; Secondary structure prediction

## 1 Introduction

Protein secondary structure (SS) is the local structure of a protein segment formed by hydrogen bonds and of great importance for studying a protein. There are two regular SS types: α-helix and β-strand, as suggested by Linus Pauling and his co-workers [1] more than 50 years ago. A protein segment connecting α-helices and/or β-strands is called coil or loop region. Instead of using only Cα atoms Kabsch and Sander [2] group SSs into eight classes with all atom coordinates. This classification is a finer-grain model of the 3-class one and contains more useful information, such as the difference between 3-helix and 4-helix. The SS content of a protein is highly correlated with its tertiary structure and function. It is also suggested that SSs play the role of basic subunits during the folding and unfolding processes of a protein [3]. Thus, SS can be considered as the transition state from primary sequence to tertiary structure. It will help in protein tertiary structure prediction and functional annotation if the SS is known [4].

Protein SS prediction from primary sequence has been an important research topic in the field of bioinformatics. It is important to predict the 8-class SS since 8-class SS provides more detailed information about the local structure of a protein. Compared with 3-class SS, 8-class can tell 3-helix and 4-helix apart and describe different types of loop regions. This



Correspondence: Professor Jinbo Xu, Toyota Technological Institute at Chicago, 6045 S Kenwood, TTIC Chicago, IL 60637, USA, jinbo.xu@gmail.com, Fax: +1-773-834-9881.

The authors have declared no conflict of interest.

detailed description of protein SS is used in solving various protein structure problems [5–7]. In the case of identifying protein conformation changing [8], using 8-class SS results in a better receiver operating characteristics curve than using 3-class. It is also much more challenging to predict the 8-class SS using machine learning methods because of the extremely unbalanced distribution of the 8-class SS types in native structures.

A variety of machine learning methods have been proposed to predict protein SS [9], especially for the 3-class (α, β or coil) SS prediction. For example, many neural network (NN) methods have been published for 3-class SS prediction [10–17]; these methods achieve Q3 accuracy of approximately 80%. PSIPRED is one of the representatives and is widely used for protein sequence analysis. However, these NN methods usually do not take the interdependency relationship among SS types of adjacent residues into consideration. Hidden Markov model (HMM) [18] is capable of describing this relationship [19, 20], but it is challenging for HMM to model the complex nonlinear relationship between input protein features and SS, especially when a large amount of heterogeneous protein features are available. Support vector machines (SVM) have also been applied for SS prediction [21–25]. Similar to the NN methods, it is also challenging for SVM to deal with the dependency among SS types of adjacent residues. In addition, SVM outputs cannot be directly interpreted as or easily transformed into likelihood/probability, which makes the prediction results difficult to interpret. Very few methods are developed for 8-class SS prediction. To the best of our knowledge, SSpro8 [26] is the only program that can predict 8-class SS of a protein. Similar to many other NN methods, SSpro8 does not exploit the interdependency relationship among SS types of adjacent residues.

In this article, we present conditional neural fields (CNFs) [27] method for 8-class protein SS prediction. CNF is a recently invented probabilistic graphical model and has been used for protein conformation sampling [28, 29] and handwriting recognition [27]. CNF is a perfect integration of CRF (conditional random fields) [29] and NNs and bears the strength of both CRFs and NNs. NNs can model the nonlinear relationship between observed protein features (e.g. sequence profiles) and SS types, but cannot easily model the interdependency among adjacent SSs. Similar to HMM, CRFs can model the interdependency among adjacent SSs, but cannot easily model the nonlinear relationship between observed protein features (e.g. sequence profiles) and SS types. By combing the strength of both NNs and CRFs, CNFs not only can model the nonlinear relationship between observed protein features and SS types, but also can model the interdependency among adjacent SSs. Similar to CRFs, CNFs can also provide a probability distribution over all the possible SS types of a given protein. That is, instead of generating a single SS type at each residue, our CNF method will generate the probability distribution of the eight SS types. The probability distribution may be useful for other purposes such as protein conformation sampling [28, 30]. Our CNF method achieves a much better Q8 accuracy than SSPro8.

Our software used in this paper is available at http://ttic.uchicago.edu/~zywang/RaptorX-SS8.

## 2 Materials and methods

We represent the input features of a given protein by an $n \times L$ matrix $X( = (\vec{X}_1, \ldots, \vec{X}_L))$, where $L$ is the number of residues in the protein. The $k$th column vector $\vec{X}_k$ represents the protein features associated with the $k$th residue. We represent the SS of a protein using a sequence of integers taking values from one to the number of SS types (i.e. 8). Formally, for a given protein with length $L$, we denote its SS as $\vec{Y}( = (Y_1, \ldots, Y_L))$, where $Y_i \in \{1, 2 \ldots, 8\}$. In Fig. 1, our model defines the conditional probability of SS $Y$ on protein features $X$ as follows:

$$P(\vec{Y}|X) \propto \exp\left(\sum_{i=1}^{L-1}\psi(Y_i, Y_{i+1}) + \sum_{i=1}^{L}\sum_{j=1}^{m}\varphi(Y_i, N_j(\vec{X}_{i-k/2}, \ldots, \vec{X}_{i+k/2}))\right) \quad (1)$$

Here, $\vec{N_j}()$ is a hidden neuron function that does nonlinear transformation of input protein features, $k$ is the window size, and $m$ is the number of hidden neuron nodes (i.e. $N_j()$). The edge feature function $\psi()$ models the interdependency between two adjacent SS types and the label feature function $\varphi()$ models the dependency of SS type on a window (with size $k$) of sequence information. Formally, $\psi()$ and $\varphi()$ are defined as follows:

$$\begin{aligned}\psi(Y_i, Y_j) &= \sum_{a,b} t_{a,b} I(Y_i = a) I(Y_j = b) \\ \varphi(Y_i, N_j) &= w_{i,j} N_j + \sum_a u_a I(Y_i = a)\end{aligned} \quad (2)$$

$I()$ is an indicator function, $w$, $u$, and $t$ are model parameters to be trained and $a$ and $b$ represent SS types. The formula in Eq. (1) is explained in Fig. 1, which contains three layers of nodes: the SS types $Y_i$, the hidden neurons $N_i$, and the input protein feature vectors $X$. The conditional probability $P(\vec{Y}|X)$ depends on $X$ and the output values from the hidden neuron nodes. Neuron nodes extract information from a shifting window of $X$. To capture higher order dependency among adjacent SS types, we combine SSs of two adjacent residues into a single second-order type, which results in the second-order CNF model. Formally, we use $\tilde{Y}_i = (Y_i, Y_{i+1})$, $i = 1, \ldots, L-1$ as the second-order type on position $i$ instead of $Y_i$. With this transformation, we obtain a second-order CNF model.

## 2.1 Model training and prediction

**2.1.1 Training—**Here, we only briefly introduce the training algorithm for CNF. Please refer to [27] for a detailed information of the CNF training algorithm. Given $K$ training proteins with features $\{U_k\} = \{(\vec{X}^k_1, \ldots, \vec{X}^k_{L_k})\}$ and their native SS types $\{V_k\} = \{(Y^k_1, \ldots, Y^k_{L_k})\}$, $k = 1, \ldots, K$, we train the CNF model by maximizing the occurring probability of the native SS types. That is, we estimate the parameters $w$ and $v$ in Eq. (2), and by maximizing $f = \prod_{k=1}^{K} P(\vec{Y}_k|X_k)$. Empirically, to avoid overfitting caused by a large number of model parameters, we add a regularization factor into the log-likelihood. That is, instead of maximizing $f$ we maximize $\log f + \gamma\|\vec{w}, \vec{t}, \vec{u}\|_2$, where $\gamma$ is a regularization factor used to control the model complexity. Although we cannot guarantee that the solution found is the optimal solution in this training problem, we can find a quietgood suboptimal solution using the LBFGS (limited memory BFGS [31]) algorithm. To obtain a good final solution, we can generate a set of suboptimal solutions with different starting solutions and use the best suboptimal solution as the final solution. The best window size $k$ and regularization factor $\gamma$ can be determined using cross-validation.

**2.1.2 Prediction—**After the model parameters are trained, we can use the model to predict the SS of a protein. We first use a forward-backward algorithm [18] to calculate the marginal probability of eight SS types at each residue, $P(Y_i|X)$. Then the SS type with the highest marginal probability can be used as the predicted SS.

## 2.2 Protein features

**2.2.1 Input features—**We use both position-dependent and position-independent protein features: PSSM (position-specific score matrix), propensity of being endpoints of SSs,

physico-chemical property, correlated contact potential of amino acids, and primary sequence. The first one feature is position-dependent and the latter four are position-independent.

PSSM contains evolutionary information derived from sequence homologs and is a position-dependent feature. The other three features are position-independent and not directly relevant to evolutionary information. To produce PSSM of a given protein, we use PSIBLAST to search it against the NR (non-redundant) database with *E*-value = 0.001 and five iterations. Low information and coiled-coil sequences in the NR database are removed as outliers using the pfilt program in the PSIPRED package.

Every amino acid has a specific propensity of being endpoints of an SS segment. Duan et al. [25] demonstrated that it helps in SS prediction by using this kind of propensity. We generated this kind of propensity by a simple statistics on a set of NR protein structures.

We also use the physico-chemical property of an amino acid as the input features. The physico-chemical property has been studied in [32] for SS prediction and is represented by a vector of seven scalars.

The matrix of correlated contact potential of amino acids estimates the correlation between two amino acids by calculating the correlation of the contact potential vectors of these two amino acids. The contact potential of an amino acid is taken from [33]. The correlated contact potential matrix has been used by Peng and Xu [34] for finding templates in protein tertiary structure prediction and has proved to be useful.

The information contained in a primary sequence can be represented by a 20 × 20 identity matrix, where each unit row vector represents an amino acid. Thus, the primary sequence is denoted as the identity matrix in the following sections.

In summary, every residue has 20-dimension PSSM features and 58-dimension position-independent features. In the experiments shown in Tables 1 and 2 and Section 3.3, we use all of those input features. The native SS of a given protein is calculated using the DSSP package [2].

**2.2.2 Neff—**Given the PSSM of a protein, we calculate a value (Neff) for each residue in the protein to evaluate the information content derived from sequence homologs

$$\text{Neff}=\exp\left(\sum_{i=1}^{20}-p_i\log p_i\right) \tag{3}$$

where $p_i$ is the frequency vector converted from PSSM for the $i$th residue in the protein. The Neff value of a given protein is calculated by the average Neff of all the residues. The Neff of a protein approximately measures the number of NR sequence homologs that can be detected by PSIBLAST from the NR database. Neff ranges from 1 to 20 since there are only 20 amino acids in protein sequences. We call those proteins with small Neff values (e.g. <5) as low-homology proteins since they do not have a large number of NR sequence homologs in the NR database. According to our experience, the more NR sequence homologs are available for a protein, the easier to predict its SS. It is very challenging to predict SS for the proteins with few sequence homologs.

### 2.3 Training and test data sets

We use two public benchmarks CB513 [16] and RS126 [13] to test the performance of our CNF method and study the relative importance of various features.

We also use a large data set cullpdb_pc30_res2.5_ R1.0_d100716 (denoted as CullPDB) from the PISCES server [35], which contains about 8000 proteins with highresolution structure (<2.5Å) and up to 30% sequence identity. We also remove protein chains with less than 50 or more than 1500 residues. For those chains with missing residues, we cut them at the missing positions into several segments. The redundancy between the CullPDB set and CB513 and RS126 is also removed.

In our results, all cross-validation we conduct is fivefold cross-validation; 4/5 of the data is used for training and 1/5 of the data for validation. In the experiments of Table 1, Supporting Information Table 1, Fig. 2 and Supporting Information Fig. 1, the results are averaged among fivefold cross-validation on their data sets. In Table 2 and Fig. 3, CNF models are trained on the data set of CullPDB. We do not do cross-validation on the data set of RS126, since it only contains 126 proteins.

## 3 Results

### 3.1 Q8 accuracy and SOV on CB513 and RS126

We use Q8 accuracy and SOV (segment OVerlap score [36]) to compare our CNF method with SSPro8 on two data sets, the CB513 data set [16] and the RS126 data set [13]. To evaluate the Q8 accuracy of SSPro8, we submit the proteins sequences to the SSpro8 server and parse results returned by the server. To evaluate the Q8 accuracy of our method, we employed two strategies. One is to conduct cross-validation on CB513. The other is to train our CNF model using the CullPDB data set and then test the CNF model on CB513 and RS126. (Redundancy between the CB513/RS126 data set and the CullPDB set is removed.) Table 1 shows the overall Q8 accuracy of our CNF method and SSPro8 and their accuracy on each SS type as well as SOV [36]. This table shows that our CNF method significantly outperforms the SSpro8 web server. Table 2 shows results on all data of CB513 and RS126 data sets. In this table, five CNF models are trained from CullPDB with different initializations and a consensus prediction is made for each residue. The accuracy on all data of CB513 is higher than the Q8 in Table 1, maybe because more data are used in training. On both CB513 and RS126, our method outperforms SSpro8 significantly.

We also conduct a cross-validation test on the CullPDB data set and achieve 67.9% Q8 accuracy in average. The confusion matrix resulted from this experiment is shown in Supporting Information Table 1. In this table, most 3/10-Helices(G) are predicted as α-Helices(H), Turns(T), and Loops(L). β-Bridges(B) are likely to be predicted as Extended strands(E) or Loops(L). Turns(T) have a high propensity to be misclassified as α-Helices(H). Bends(S) are probably to be predicted as Loops(L) and Turns(T). There are also another two types of confusions. One is the confusion inside the same type of SSs, such as H, G are both helices, and E, B are both strands. The other is overlap between different classes. A Turn(T) is defined in [2] as one amino acid has a hydrogen bond with another, but not in a helix. This definition implies the similarity between a Turn and a Helix.

### 3.2 Relative importance of various features

There is no doubt that the PSSM is the most important information for SS prediction. Here, we examine the relative importance of various position-independent features for SS prediction. Our analysis shows that the propensity of being SS endpoints contributes more than other features.

We compare those Q8 accuracy values of four position-independent features (physico-chemical property, propensity of SS endpoints, correlated contact potentials, and identity matrix) without PSSM features in Supporting Information Fig. 1. No matter what the regularization factor is, the feature of SS endpoints works better than other features.

However, when PSSM is also used, there seems to be no obvious difference among these position-independent features. Figure 2 shows Q8 accuracy of our CNF method with different features and trained with different regularization factors. Among the highest accuracy of all the CNF models, the gap is <1% in Q8 accuracy. In this figure, Test F-1 uses merely PSSM features, Test F-2 uses PSSM and Neff, Test F-3 uses PSSM and physico-chemical features, Test F-4 uses PSSM and SS endpoints, Test F-5 uses PSSM and contacts potentials, and Test F-6 uses PSSM and identity matrix. Every accuracy value is the average of five repeated experiments with same model parameters. The Q8 accuracy values in Fig. 2 of different features reach their maximum with different regularization factors, but are not far from each other, which indicates that position-independent features do not help improve 8-class SS prediction if PSSM is used.

### 3.3 Q8 accuracy on proteins with various Neff values

We also examine the performance of our method with respect to the Neff value of a protein. As shown in Fig. 3, we divide the Neff values into nine intervals and calculate Q8 accuracy of both our CNF method and SSPro8 on CB513 with respect to Neff. On average our method performs better than SSPro8 no matter which Neff interval is considered. Our CNF model performs best on the proteins with Neff values in [7,8). The performance of our CNF method increases with respect to Neff when it is <8. It is interesting to notice that the performance decreases when Neff is larger than 8. This may imply that for SS prediction, it may not be the best strategy to use evolutionary information in as many homologs as possible. Instead, we should use a subset of sequence homologs to build sequence profile when there are many sequence homologs available. That is, we should not use a sequence profile with Neff larger than eight to predict SS. One possible method to reduce Neff is to run PSIBLAST with smaller *E*-value or fewer iterations.

## 4 Discussions

We have presented a method for 8-class SS prediction using CNFs. Our CNF model not only captures the complex nonlinear relationship between protein features and SS, but also exploits interdependence among SS types of adjacent residues. This is why we can achieve a much better performance than the existing methods. Furthermore, our CNF model defines a probability distribution over the SS space. The probability distribution provided by our method can be in turn applied to protein conformation sampling [28, 30].

The error in SS prediction may result from various factors. Similar to most existing methods, our method does not take into consideration long-range inter-residue interaction, which may be important for β-sheet prediction. This may be the major reason why we cannot do prediction on the β-strand as well as the α helix. The unbalanced distribution of SS types also makes it challenging to predict 8-class SS, especially for 3/10-helix, β-bridge and π-helix. These SS types in total account for only around 5% of residues.

Our method also achieves Q3 accuracy 81.17% on the CullPDB data set, comparable to the popular three-class SS prediction program PSIPRED. We can further improve our method by improving the wiring scheme used to connect input features to hidden neurons so as to extract more information from sequence profiles and the chemical property profile of amino acids. It may also help by increasing the number of output states in our CNF model. As discussed in this paper, amino acids have different propensities of being in the two ends of a single SS segment. Therefore, we can split an SS state into several subcategories, like the head of a helix, the tail of a helix, and the middle of a helix.

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## Acknowledgments

This work is financially supported by the National Institutes of Health grant R01GM089753 (to J. X.) and the National Science Foundation grant DBI-0960390 (to J. X.). The authors are also grateful to TeraGrid for their computational resources through grants TG-MCB100062 and TG-CCR100005 (to J. X.).

## Abbreviation

| | |
|---|---|
| **CNFs** | conditional neural fields |
| **CRF** | conditional random field |
| **HMM** | hidden Markov model |
| **NN** | neural network |
| **NR** | non-redundant |
| **PSSM** | position-specific score matrix |
| **SOV** | segment overlap score |
| **SS** | secondary structure |
| **SVM** | support vector machines |

## References

1. Pauling L, Corey RB, Branson HR. Two hydrogen-bonded helical configurations of the polypeptide chain. Proc Natl Acad Sci USA. 1951; 37:205–211. [PubMed: 14816373]
2. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 1983; 22:2577–2637. [PubMed: 6667333]
3. Karplus M, Weaver DL. Protein folding dynamics: the diffusion–collision model and experimental data. Protein Sci. 1994; 3:650–668. [PubMed: 8003983]
4. Myers JK, Oas TG. Preorganized secondary structure as an important determinant of fast protein folding. Nat Struct Mol Biol. 2001; 8:552–558.
5. DeBartolo J, Colubri A, Jha AK, Fitzgerald JE, et al. Mimicking the folding pathway to improve homology-free protein structure prediction. Proc Natl Acad Sci USA. 2009; 106:3734–3739. [PubMed: 19237560]
6. Boden M, Yuan Z, Bailey T. Prediction of protein continuum secondary structure with probabilistic models based on NMR solved structures. BMC Bioinformatics. 2006; 7:68. [PubMed: 16478545]
7. Cheng J, Baldi P. Improved residue contact prediction using support vector machines and a large feature set. BMC Bioinformatics. 2007; 8:113. [PubMed: 17407573]
8. Bodén M, Bailey TL. Identifying sequence regions undergoing conformational change via predicted continuum secondary structure. Bioinformatics. 2006; 22:1809–1814. [PubMed: 16720586]
9. Pirovano W, Heringa J. Protein secondary structure prediction. Data Mining Tech Life Sci. 2010; 609:327–348.
10. Qian N, Sejnowski TJ. Predicting the secondary structure of globular proteins using neural network models. J Mol Biol. 1988; 202:865–884. [PubMed: 3172241]
11. Holley LH, Karplus M. Protein secondary structure prediction with a neural network. Proc Natl Acad Sci USA. 1989; 86:152–156. [PubMed: 2911565]
12. Kneller DG, Cohen FE, Langridge R. Improvements in protein secondary structure prediction by an enhanced neural network. J Mol Biol. 1990; 214:171–182. [PubMed: 2370661]

13. Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol. 1993; 232:584–599. [PubMed: 8345525]
14. Rost B, Sander C. Combining evolutionary information and neural networks to predict protein secondary structure. Proteins. 1994; 19:55–72. [PubMed: 8066087]
15. Rost B. PHD: predicting one-dimensional protein structure by profile-based neural networks. Methods Enzymol. 1996; 266:525–539. [PubMed: 8743704]
16. Cuff JA, Barton GJ. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins. 1999; 34:508–519. [PubMed: 10081963]
17. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol. 1999; 292:195–202. [PubMed: 10493868]
18. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE. 1989:275–286.
19. Asai K, Hayamizu S, Handa KI. Prediction of protein secondary structure by the hidden Markov model. Bioinformatics. 1993; 9:141–146.
20. Aydin Z, Altunbasak Y, Borodovsky M. Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. BMC Bioinformatics. 2006; 7:178. [PubMed: 16571137]
21. Hua S, Sun Z. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol. 2001; 308:397–407. [PubMed: 11327775]
22. Kim H, Park H. Protein secondary structure prediction based on an improved support vector machines approach. Protein Eng. 2003; 16:553–560. [PubMed: 12968073]
23. Ward JJ, McGuffin LJ, Buxton BF, Jones DT. Secondary structure prediction with support vector machines. Bioinformatics. 2003; 19:1650–1655. [PubMed: 12967961]
24. Guo J, Chen H, Sun Z, Lin Y. A novel method for protein secondary structure prediction using dual-layer SVM and profiles. Proteins. 2004; 54:738–743. [PubMed: 14997569]
25. Duan M, Huang M, Ma C, Li L, Zhou Y. Position-specific residue preference features around the ends of helices and strands and a novel strategy for the prediction of secondary structures. Protein Sci. 2008; 17:1505–1512. [PubMed: 18519808]
26. Pollastri G, Przybylski D, Rost B, Baldi P. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins. 2002; 47:228–235. [PubMed: 11933069]
27. Peng, J.; Bo, L.; Xu, J. Advances in Neural Information Processing Systems. Bengio, Y.; Schuurmans, D.; Lafferty, J.; Williams, CKI.; Culotta, A., editors. Vol. 22. MIT Press; 2009. p. 1419-1427.
28. Zhao, F.; Peng, J.; DeBartolo, J.; Freed, K., et al. Research in Computational Molecular Biology. Batzoglou, S., editor. Springer; Berlin/Heidelberg: 2009. p. 59-73.
29. Lafferty, JD.; McCallum, A.; Pereira, FCN. Proceeding of International Conference on Machine Learning. Morgan Kaufmann Publishers Inc; San Francisco, CA, USA: 2001. p. 282-289.
30. Zhao F, Li S, Sterner BW, Xu J. Discriminative learning for protein conformation sampling. Proteins. 2008; 73:228–240. [PubMed: 18412258]
31. Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Progr. 1989; 45:503–528.
32. Meiler J, Müller M, Zeidler A, Schmäschke F. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. J Mol Model. 2001; 7:360–369.
33. Tan YH, Huang H, Kihara D. Statistical potential-based amino acid similarity matrices for aligning distantly related protein sequences. Proteins. 2006; 64:587–600. [PubMed: 16799934]
34. Peng J, Xu J. Low-homology protein threading. Bioinformatics. 2010; 26:294.
35. Wang G, Dunbrack RLJ. PISCES: a protein sequence culling server. Bioinformatics. 2003; 19:1589–1591. [PubMed: 12912846]
36. Rost B, Sander C, Schneider R. Redefining the goals of protein secondary structure prediction. J Mol Biol. 1994; 235:13–26. [PubMed: 8289237]

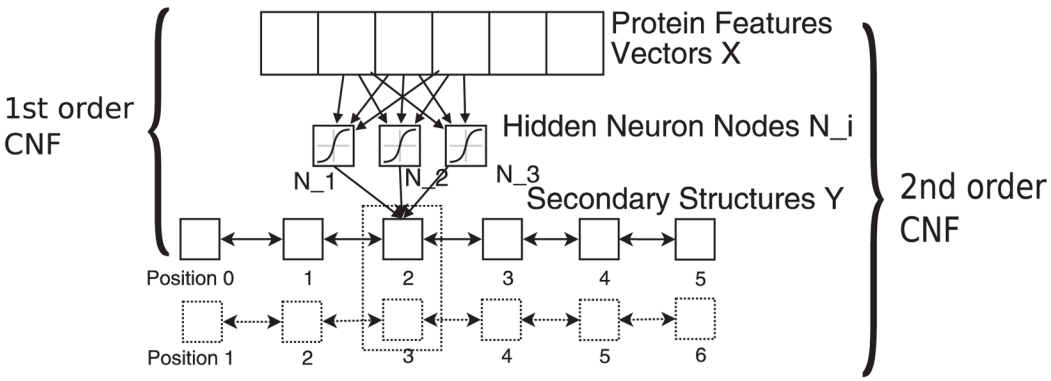


**Figure 1.**

The CNF model for 8-class protein SS prediction. CNFs model the relationship between $X$ and $Y$ through a hidden layer of neuron nodes, which conduct nonlinear transformation of $X$. The second-order CNF model combines SSs of two adjacent positions into a single second-order type. For example, the dash box shows a combined second-order type formed by the SS types at positions 2 and 3.

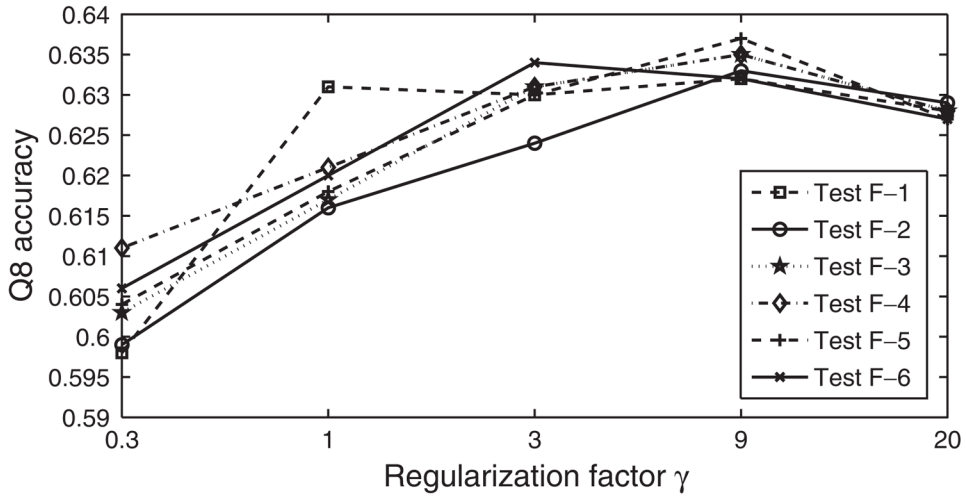


**Figure 2.**

Q8 accuracy of our CNF method on the data set of CB513 with different position-independent features combined with PSSM features and trained with different regularization factors.

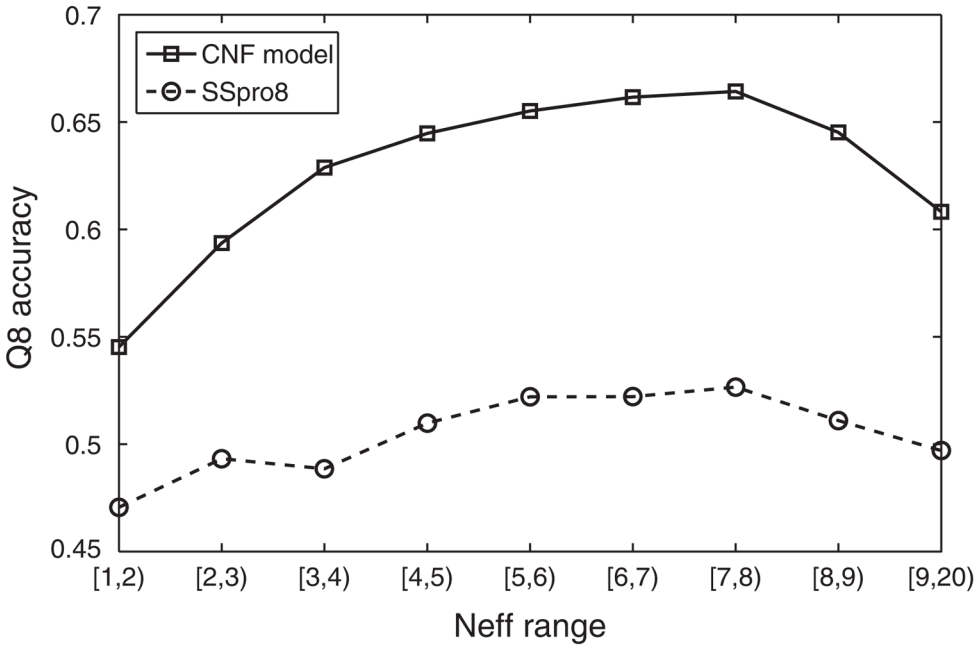


**Figure 3.**
Q8 accuracy on CB513 within respect to Neff. Each point represents an average Q8 accuracy on those proteins within a Neff interval. The solid and dash lines correspond to our CNF model and SSpro8, respectively.

## Table 1

Q8 accuracy of our CNF method and SSPRO8 on the CB513 data set

| | CNF method | | SSpro8 | |
|---|---|---|---|---|
| | **Mean** | **Std** | **Mean** | **Std** |
| Q8 | **0.633** | 0.013 | 0.511 | 0.015 |
| Q-H | **0.887** | 0.009 | 0.752 | 0.031 |
| Q-G | **0.049** | 0.015 | 0.007 | 0.002 |
| Q-I | 0 | 0 | 0 | 0 |
| Q-E | **0.776** | 0.016 | 0.597 | 0.013 |
| Q-B | 0 | 0 | 0 | 0 |
| Q-T | **0.418** | 0.008 | 0.327 | 0.017 |
| Q-S | **0.117** | 0.015 | 0.049 | 0.003 |
| Q-L | **0.608** | 0.014 | 0.499 | 0.019 |
| SOV | **0.206** | 0.025 | 0.141 | 0.015 |

Q-H, Q-G, Q-I, Q-E, Q-B, Q-T, Q-S, and Q-L represent the prediction accuracy on a single SS type H, G, I, E, B, T, S, and L respectively. Bold indicates improvement.

**Table 2**

The average Q8 accuracy and the sensitivity of each subclass tested on CB513 and RS126 using five CNF models trained from the CULLPDB data

| | On all data of CB513 | | | On all data of RS126 | | |
|---|---|---|---|---|---|---|
| | CNF method | | SSpro8 | CNF method | | SSpro8 |
| | Mean | Std | | Mean | Std | |
| Q8 | **0.649** | 0.003 | 0.51 | **0.647** | 0.003 | 0.48 |
| Q-H | **0.875** | 0.004 | 0.752 | **0.9** | 0.005 | 0.728 |
| Q-G | **0.207** | 0.021 | 0.006 | **0.229** | 0.022 | 0.016 |
| Q-I | 0 | 0 | 0 | 0 | 0 | 0 |
| Q-E | **0.756** | 0.001 | 0.598 | **0.797** | 0.002 | 0.577 |
| Q-B | 0 | 0 | 0 | 0 | 0 | 0 |
| Q-T | **0.487** | 0.004 | 0.33 | **0.488** | 0.01 | 0.308 |
| Q-S | **0.202** | 0.009 | 0.051 | **0.153** | 0.009 | 0.051 |
| Q-L | **0.601** | 0.002 | 0.5 | **0.614** | 0.006 | 0.479 |
| SOV | **0.194** | **0.002** | **0.143** | **0.18** | **0.003** | **0.123** |

Q-H, G, I, E, B, T, S, L are the same with Table 1. Bold indicates improvement.

*Proteomics*. Author manuscript; available in PMC 2012 May 02.