Improving Protein Secondary-Structure Prediction by Predicting Ends of Secondary-Structure Segments

Uros Midic[1] A. Keith Dunker[2] Zoran Obradovic[1*]

[1]Center for Information Science and Technology
Temple University
1805 N. Broad St., 303 Wachman Hall
Philadelphia, PA 19129 USA

[2]Center for Computational Biology and Bioinformatics
Indiana University School of Medicine
714 North Senate Avenue, Suite 250
Indianapolis, IN 46202 USA

Abstract – **Motivated by known preferences for certain amino acids in positions around a-helices, we developed neural network-based predictors of both N and C a-helix ends, which achieved about 88% accuracy. We applied a similar approach for predicting the ends of three types of secondary structure segments. The predictors for the ends of H, E and C segments were then used to create input for protein secondary-structure prediction. By incorporating this new type of input, we significantly improved the basic one-stage predictor of protein secondary structure in terms of both per-residue (Q_3) accuracy (+0.8%) and segment overlap (SOV_3) measure (+1.4).**

I. Introduction

Prediction of secondary structure from amino acid sequence is useful for developing and testing structure-function hypotheses, for improving sequence alignments of remote homologues, for improving homology models of sequences with low sequence identity compared to the template molecule, and as the starting point for prediction of 3-D structure [1,2]. Secondary structure prediction is mature problem that may be reaching the upper bound of the accuracy that can be achieved [3].

Most current state-of-the-art methods use data obtained from sequence alignment profiles to classify each residue as one of three general structure types, helix – H, sheet – E and coil – C [3-7]. These methods now achieve near 78% Q_3 accuracy [8]. Another very important measure of prediction quality is "segment overlap" (SOV_3) measure [9]. SOV_3 measures how similar the distribution of predicted segments is to the distribution of actual segments, as well as how close the ends of the overlapping predicted segment are to the ends of the actual segment.

Empirical evidence has been presented indicating preferences of certain amino acids at specific positions around the ends of a-helices [10-12] and also indicating preferences for positions throughout the entire lengths of a-helices [13]. The idea that structural boundaries are accompanied by particular amino acid biases motivated us to attempt to develop a predictor of the boundaries between structured and intrinsically unstructured regions in proteins. This boundary predictor was fairly successful and led to modest improvements in predicting intrinsically unstructured regions in proteins [14].

The success on structured/unstructured boundary prediction encouraged us to consider the possibility of predicting the ends of a-helices; the latter problem has the advantage of a much larger set of data compared to the former. Here we test a-helix/non-a-helix boundary prediction using neural networks. We also test whether the ends of other structural types are predictable. To the best of our knowledge, no significant results regarding this problem have been reported to date. Finally, we test whether prediction of ends (or boundaries) can be used to improve overall secondary structure predictions.

Based on neural networks, we built predictors for the N-ends and C-ends of a-helices from the protein sequence that achieve true accuracy of 88.2% and 88.5%, respectively. The importance of correctly predicted boundaries between secondary-structure segments, coupled with the successful prediction of a-helix ends motivated us to generalize the prediction of a-helix ends and predict the N- and C- ends of segments of types H, E and C with six predictors – two for each segment type. We assumed that if these predictors were fairly successful, their outputs could be used as input to predict secondary structure. The rationale for this assumption is simple. Assuming we were given perfect predictors (i.e. with 100% accuracy) for these 6 problems and applied them on some protein sequence, we would be able to make a perfect prediction of secondary structure by using a deterministic algorithm. For each residue, we would have to find the nearest residue that is predicted to be the N-end and the nearest residue that is predicted to be the C-end of a secondary-structure segment. For perfect segment-end predictors, the nearest N-end residue and nearest C-end residue would be of the same type, so observed residue should be classified as belonging to that type. Furthermore, it is easy to prove that only four out of the six perfect predictors are sufficient for perfect secondary-structure prediction (given that they are perfect and that we throw out a pair of N-end and C-end predictors of the same segment type). Although predictors for the six segment-ends problems cannot be perfect, we still hoped that we would be able to use them to improve secondary structure prediction.

Our goal was to compare predictions based on data, which was obtained from sequence alignment profiles, and predic-

*) Correspondence: E-mail: zoran@ist.temple.edu. Phone: +215 204 6265. Fax: +215 204 5082.

0-7803-9387-2/05/$20.00 ©2005 IEEE

tions based on output from segment-ends predictors. We used the same simple prediction model based on neural networks. We also tried several methods of combining two types of data to achieve even better prediction accuracy.

II. METHODS

A. Dataset

The data source was a non-redundant set of 4460 protein chains with a known 3D structure (resolution<2.5 and R-factor<0.25). All pairs of chains in the set have less than 30% pairwise sequence identity [15].

Several methods have been used for encoding amino acid residues as numerical values.

Binary (or *sparse*) representation is a vector of 20 binary numbers assigned to each residue. One of these 20 numbers (depending on the type of amino acid) is 1 and the remaining 19 numbers are 0. Datasets produced by using binary encoding are very sparse, with only 5% of nonzero values. This affects the process of training, due to the "curse of dimensionality" problem.

Substitution matrix representation (sometimes called *PAM encoding* or *BLOSUM encoding*) encodes each residue as a corresponding row of 20 numbers from a substitution matrix, e.g. BLOSUM62. This improves generalization ability of a predictor, since similar amino acid residues are represented with similar vectors.

Sequence profile representation encodes each residue as a vector of 20 substitution scores from a sequence profile. This further improves generalization ability of a predictor, since sequence profiles include additional evolutionary information. We used it for both of our problems, prediction of secondary structure and prediction of ends of secondary-structure segments.

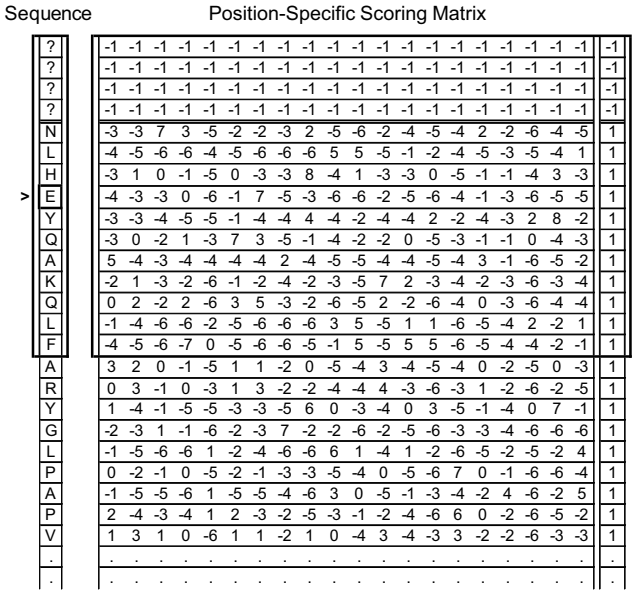

Fig. 1. 300 input attributes that encode for one residue are rows taken from the position-specific scoring matrix for a window of 15 residues. Additional 15 attributes indicate when residue positions are out of sequence bounds.

The substitution scores were taken from a position-specific scoring matrix (PSSM) generated by PSI-BLAST [16]. PSI-BLAST was run for three iterations, using the *nr* database, the BLOSUM62 substitution matrix and an *e-value* threshold of 0.0001. PSSM generated by PSI-BLAST contains rows of 20 log-odds values, one for each residue. We constructed 300 input attributes that represent a residue by joining rows from PSSM for a window of 15 residues with the observed residue in the center of the window (Fig. 1). If some position in the window extended beyond an end of the sequence, we inserted BLOSUM62 substitution scores for X, i.e. unknown residue. This scheme produces 15*20 = 300 input attributes. We added 15 additional input attributes, each of which had a value 1 if the corresponding residue is known or a value -1 if the corresponding residue is unknown, i.e. that position in a window falls outside of a sequence. All 315 input attributes were normalized to obtain the Gaussian distribution.

We used the DSSP assignment method [17] to obtain an 8-state secondary-structure assignment from PDB files. These 8 states – *G, H, I, T, E, B, S and _ (i.e. blank)* – can be further reduced to three more general states – **H** (helix), **E** (sheet) and **C** (coil) – using various schemes. The choice of the scheme can affect the prediction accuracy [3]. We chose the scheme used by the EVA server [18], to assign *H, G* and *I* to **H**; *E* and *B* to **E**; and *S, T* and _ to **C**. For the remainder of this paper, we will simply use H, E and C to denote the three general states of secondary structure.

B. Model, training and prediction

We chose a neural network with a single hidden-layer as our prediction model. Neural networks were trained using standard back-propagation [19] (for prediction of ends of segments) or resilient propagation [20] (for prediction of secondary structure). For secondary-structure problem, we trained three separate networks. All networks used the same input. Each of them had a separate (smaller) hidden layer and produced one of three outputs (Fig. 2). Three networks with the same input and separate hidden layers and outputs can still be observed as a single composite neural network with three outputs. The corresponding composite neural network has a hidden layer that is three-times larger, where two-thirds of the

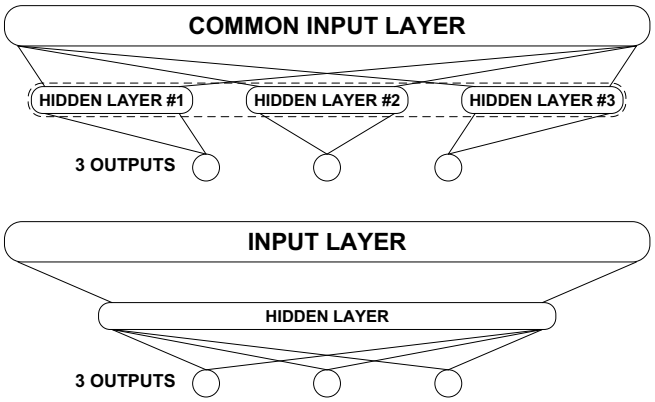

Fig. 2. Three neural networks with common input and separate hidden layers, interpreted as one composite neural network (top), are used instead of a regular neural network with one hidden layer (bottom).

weights and biases have zero value. Training of separate networks was necessary due to the high memory requirements of the resilient propagation algorithm. Tests on smaller-scale networks showed that the results obtained by the composite neural networks were comparable to regular neural networks.

The dataset was divided into 16 disjoint subsets, each containing an approximately equal number of residues. Residues from one protein chain were always grouped in one subset to enable the evaluation of SOV_3 (introduced in section *G*). For each of the problems, we trained 16 separate predictors, one using each subset. We then integrated those separately trained predictors into one ensemble predictor by averaging their outputs. E.g. when prediction is tested on one of the 16 subsets, we use an ensemble predictor consisting of 15 predictors trained on the other 15 subsets. For the remainder of this paper, we will use "predictor" for this ensemble predictor and "single predictor" or "component predictor" for predictors that are components of the ensemble predictor.

C. Prediction of the ends of a-helices

We repeated the analysis (from [10]) of position-specific amino acid propensities around the ends of a-helices on our larger set of proteins. Position-specific (relative) propensity for amino acid *j* (AA_j) at position *i* is defined as $Pr_i(AA_j) = Fr_i(AA_j)/Fr(AA_j)$, i.e. the ratio of the frequency of AA_j at position *i* over the overall frequency of AA_j. If $Pr_i(AA_j)$ is significantly larger than 1, there is a positive preference for amino acid *j* in position *i*. Conversely, if $Pr_i(AA_j)$ is significantly smaller than 1, there is a negative preference for amino acid *j* in position *i*. Since certain propensities become more emphasized when short a-helices are filtered out, we decided to keep only a-helices with at least 8 residues.

Every a-helix has two ends, the N-end and the C-end. We define two problems: the prediction of N-ends of a-helices (labeled N_A) and the prediction of C-ends of a-helices (labeled C_A). We composed the dataset for N_A problem by including all residues at the N-ends of a-helices (with at least 8 residues) as positive instances, and an equal number of randomly selected remaining residues as negative instances. Each residue is assigned 315 input attributes (as discussed above). Instance class was encoded as a numerical pair, (0.9, 0.1) for positive instances and (0.1, 0.9) for negative instances. The dataset for the C_A problem was constructed using the same approach.

We used neural networks with 20 hidden neurons and two outputs, and trained them using standard back-propagation algorithm, with learning rate 0.2 and momentum 0.8. Part of the training set (20%) was used as a validation set to avoid over-fitting. During prediction, two output values are compared and the greater value decides how the observed residue will be classified. We will label the two predictors as PNA (N_A problem) and PCA (C_A problem)

D. Prediction of the ends of 3-state secondary structure segments

The problem of predicting the ends of a-helices can be generalized to any other type of sequence segment. We experimented with three types of secondary structure segments (reduced to 3-state from DSSP's 8-state assignment): helix (H), sheet (E) and coil (C). For each of these three types of segments, we define two distinct problems: prediction of the N-ends and prediction of the C-ends. Overall we have 3*2 = 6 distinct problems, that we will label as *<segment-end type>-<segment type>, i.e.* N_H, C_H, N_E, C_E, N_C and C_C.

Datasets for these 6 problems were constructed in a way similar to the construction of datasets for N_A and C_A. The main difference was that we did not perform any filtering of short segments. Note that a-helices are not equivalent with H segments. To be more precise, every a-helix is part of an H segment that may be equivalent to the a-helix but can also include some 3_{10} and p-helix residues. A residue that is located at an end of the a-helix may lie in the middle of an H segment. Therefore, problems N_H and C_H are not identical to N_A and C_A.

We will label six predictors for problems N_H, C_H, N_E, C_E, N_C and C_C as PNH, PCH, PNE, PCE, PNC and PCC, respectively. We used the same neural network architecture and training procedure as for PNA and PCA predictors. Due to more heterogeneous character of H, E and C segments, we expected a lower prediction accuracy than was obtained for N_A and C_A problems.

E. Prediction of secondary structure using inputs obtained from sequence profiles

Many existing predictors of secondary structure use a neural network with a single hidden layer and profile score data as input. Most of predictors also filter their results, usually by using a second stage of prediction.

Our basic predictor of secondary structure (labeled PSS_{PROF}) uses 315 inputs obtained from the sequence alignment profiles. The type of 3-state secondary structure assigned to each residue was encoded as a triplet of numbers: (0.9, 0.1, 0.1) for H, (0.1, 0.9, 0.1) for E and (0.1, 0.1, 0.9) for C. As already discussed, each individual predictor consisted of three neural networks that were trained separately, but we observe them together as one composite neural network. During the prediction phase, the largest of the three output values decides how the observed residue will be classified. Neural networks had 50 hidden neurons and were trained using resilient propagation.

F. Prediction of secondary structure using the output of segment-ends predictors

To test whether outputs of predictors for problems N_H, C_H, N_E, C_E, N_C and C_C could be used as input to predict secondary structure, we first submitted the protein sequences to six segment-ends predictors (PNH, PCH, PNE, PCE, PNC and PCC). The result for each sequence was a matrix of L*6 values, where L is the length of sequence. We then constructed 90 new input attributes for each residue by joining rows from this matrix for a window of 15 residues with observed residue in the center of the window (Fig. 3). If a position in the window extended beyond an end of the sequence, we inserted zeros (because such a position cannot be at an end of an H, E or C segment).

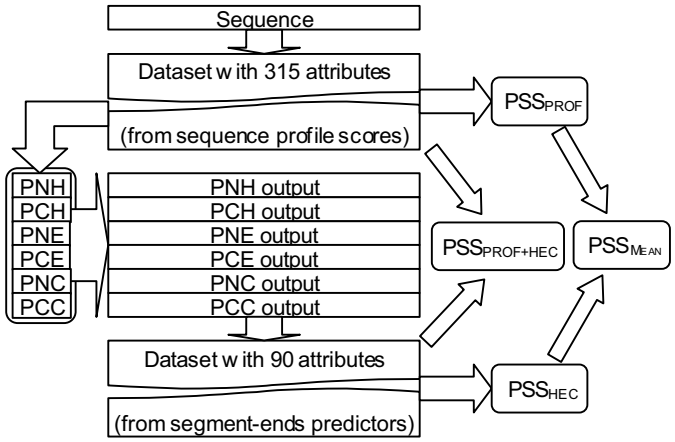

Fig. 3. The flowchart of data from sequence to secondary-structure predictors.

We used this new data (90 input attributes) – instead of the previously described 315 input attributes – as input for predictor of secondary structure (labeled PSS_{HEC}). We used the same model, training and prediction techniques used for PSS_{PROF}.

We first examined the effects of various sizes of the input window by using subsets of the set of input attributes. We trained and evaluated predictors with input window sizes of 7, 9, 11, 13 and 15 residues (labeled $PSS_{HEC;7}$, $PSS_{HEC;9}$, etc.), i.e. using 42, 54, 66, 78 and 90 input attributes.

We then compared the importance of attributes coming from six segment-ends predictors by excluding attributes obtained from one pair of predictors for the same segment type, and using only the attributes obtained from the remaining two pairs of predictors. Window size was set to 15 residues, so the number of input attributes was always 60 (out of 90 possible). We label these three predictors PSS_{HC} (uses only data obtained form PNH, PCH, PNC and PCC), PSS_{HE} and PSS_{EC}.

Performance of the PSS_{HEC} predictor was then compared to the performance of PSS_{PROF} predictor. Both predictors used the same input window size (15 residues) and had the same size of hidden layers in neural networks (50 neurons).

We took two approaches for using sequence profile data and output of the segment-ends predictors together. The first approach was to train another predictor (labeled $PSS_{PROF+HEC}$) that used 405 input attributes, 315 attributes coming from sequence profile data and 90 from predictors of segment ends. Once again we used the same input window and hidden layers sizes.

The second approach was to use PSS_{PROF} and PSS_{HEC} separately, obtain two sets of outputs, and use means of pairs of values from those two sets to make the final prediction. We label this predictor as PSS_{MEAN}.

G. Second stage of secondary-structure prediction

Many secondary-structure prediction methods involve two stages of prediction [4,5,21]. The stage-two predictor usually takes the outputs from the main (stage-one) predictor over a window of residues around the observed residue. The role of the stage-two predictor is to improve accuracy by filtering output from the stage-one predictor.

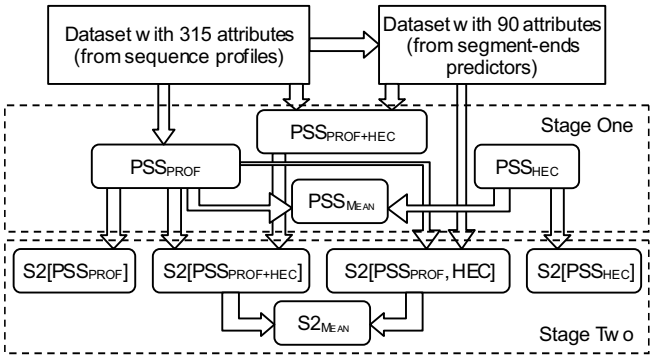

Fig. 4. Stage two of secondary-structure prediction.

We used a window of 15 residues to create the input for stage-two predictors from outputs of main predictors. All stage-two predictors were based on neural networks with 50 hidden neurons and were trained using resilient propagation.

Basic stage-two predictor $S2[PSS_{PROF}]$ uses output from PSS_{PROF}. Predictor $S2[PSS_{HEC}]$ uses output from PSS_{HEC}, while $S2[PSS_{PROF+HEC}]$ uses output from $PSS_{PROF+HEC}$. We also trained a second-stage predictor with "hybrid" input: $S2[PSS_{PROF}, HEC]$ uses output from stage-one predictor PSS_{PROF} and direct output from predictors of segment-ends (Fig 4.). Similar to PSS_{MEAN}, stage-two predictor $S2_{MEAN}$ takes outputs from $S2[PSS_{PROF+HEC}]$ and $S2[PSS_{PROF}, HEC]$, and calculates means of pairs of values from those two sets to make the final prediction.

H. Prediction & Evaluation Procedure

Cross-validation was performed by testing the predictor – which is an ensemble of single predictors trained on 15 separate subsets – using the remaining subset as a test set. This procedure was repeated 16 times, where each time a different subset was used as a test set. All reported results are average over those 16 tests. Note that the standard deviation was calculated over all tests and not over individual proteins. Its purpose is solely to measure the stability of prediction methods and not to estimate the distribution of values of performance indicators when they are measured on individual proteins.

We also tested 16 single predictors individually to check prediction improvements when single predictors are integrated into an ensemble. We will not report these results, except in a few interesting cases.

Predictors of segment ends were evaluated using four measures. $\textit{Sensitivity} = \frac{\text{num. of correctly predicted positives}}{\text{number of all positives}}$ measures the accuracy of a predictor on the subset of positive instances, while $\textit{specificity} = \frac{\text{num. of correctly predicted negatives}}{\text{number of all negatives}}$ measures the accuracy of a predictor on the subset of negative instances. $\textit{True accuracy} = \frac{\text{sensitivity+specificity}}{2}$ is preferred over plain accuracy (fraction of correctly predicted instances) for binary classification problems, where one class dominates the set of possible instances. (E.g. the number of residues that are not at an N-end of a helix segment is much larger than the number of residues that are at N-ends of helix segments.)

The fourth measure is the *integral of the ROC curve*. Our predictors classify instances by comparing two output values. However, depending on application, it might be appropriate to add a parameter *?* to one of the two outputs, and thus introduce positive (when *?* > 0) or negative (when *?* < 0) bias towards the class represented by that output. The ROC curve is a plot obtained by changing the value of ? from -1 to 1, measuring *specificity* and *sensitivity* of the biased predictor and plotting *specificity* against *sensitivity*. The area beneath the ROC curve (integral on interval [0, 1]) is a measure of the quality of a predictor. This measure will be 1 for a perfect predictor and 0.5 for a "random" predictor. If a predictor is biased towards a class that is dominant in a distribution (e.g. negative class in the case of our segment-ends problems), it can have a very good plain accuracy because it correctly predicts many instances in the dominant class, but a very low ROC curve integral. Note that two predictors can have a similar true accuracy, but dissimilar ROC curve integrals. A high value of the ROC curve integral means that the predictor can be finely tuned by changing parameter ?, e.g. to minimize false negative predictions while retaining good sensitivity.

We evaluated predictors of secondary structure using Q_3 accuracy (overall fraction of residues with correctly predicted 3-state secondary structure), $Q_{3\text{-}H}$ accuracy (fraction of correctly predicted helix residues), $Q_{3\text{-}E}$ accuracy (fraction of correctly predicted sheet residues) , $Q_{3\text{-}C}$ accuracy (fraction of correctly predicted coil residues), and SOV_3 measure (as defined in [9]).

TABLE I
POSITION-SPECIFIC AMINO ACID PROPENSITIES NEAR N-ENDS OF A-HELICES; N1 IS THE FIRST RESIDUE INSIDE HELIX, FOLLOWED BY N2, N3, ETC.; NCAP IS THE FIRST RESIDUE OUTSIDE HELIX, FOLLOWED BY N', N? AND N?'; VALUES LARGER THAN 1.5 ARE **BOLD**; SMALLER THAN 0.67 UNDERLINED

	Outside helix				Inside helix					
	N?'	N?	N'	Ncap	N1	N2	N3	N4	N5	N6
A	0.93	0.93	0.80	0.43	1.07	1.20	1.17	1.49	**1.52**	**1.74**
C	0.90	0.93	0.77	0.84	0.47	0.49	0.64	0.92	0.83	0.74
D	1.04	1.02	0.99	**2.61**	0.84	**1.57**	**1.54**	0.37	0.84	0.84
E	0.94	1.01	0.80	0.60	1.34	**2.60**	**2.21**	0.53	1.41	**1.57**
F	0.89	0.95	1.01	0.49	0.99	0.58	0.80	1.45	0.89	0.81
G	1.24	**1.62**	1.20	1.26	0.69	0.78	0.61	0.37	0.37	0.38
H	1.12	1.13	0.92	1.16	0.78	0.92	0.99	0.71	0.78	0.80
I	0.82	0.74	1.06	0.26	0.81	0.52	0.69	**1.79**	1.14	0.79
K	0.97	0.94	0.88	0.55	1.05	1.10	0.83	0.85	1.34	**1.53**
L	0.87	0.74	1.22	0.33	0.97	0.55	0.84	**1.74**	1.30	1.02
M	1.00	0.76	**1.52**	0.43	1.00	0.65	0.89	**1.63**	1.22	0.97
N	1.03	1.09	1.06	**2.38**	0.55	0.88	0.63	0.47	0.82	0.86
P	1.30	1.42	1.21	1.24	**2.72**	1.04	0.59	0.00	0.00	0.24
Q	1.00	0.86	0.82	0.60	1.00	1.26	**1.83**	1.15	1.41	**1.59**
R	1.06	0.84	0.83	0.58	1.00	0.95	0.74	1.36	1.38	**1.61**
S	1.13	1.12	1.02	**2.67**	0.80	1.07	0.78	0.50	0.70	0.76
T	1.00	1.07	1.03	**2.15**	0.80	0.85	1.13	0.70	0.74	0.77
V	0.88	0.83	0.99	0.24	0.86	0.57	0.90	**1.51**	1.00	0.67
W	0.83	0.81	0.94	0.45	1.25	0.83	0.71	1.21	0.90	0.88
Y	1.02	0.92	0.88	0.58	0.91	0.65	0.84	1.07	0.80	0.80

III. RESULTS & DISCUSSION

Prediction of the ends of a-helices and ends of 3-state secondary structure segments

Position-specific amino acid propensities for positions around the ends of a-helices are listed in Tables I and II. Values larger than 1.5 are printed in bold (positive preference), while values smaller than 1/1.5=0.67 are underlined (negative preference). There is a high level of consistency with previously published findings. Since we used the DSSP method for secondary structure assignment, some of the propensities are less emphasized than in studies that use assignment methods based on the geometry of a protein's backbone (e.g. preference for Glycine in position Ccap is lower while preference for Glycine in neighboring position C' is higher). On the other hand, some propensities are more emphasized, due to the larger set of protein chains (e.g. preference for Proline in positions C'-C?').

Evaluation results for predicting the ends of a-helices (problems N_A and C_A) and segments of types H, E and C (problems N_H, C_H, N_E, C_E, N_C and C_C) are listed in Table III. The performance of predictors PNA and PCA was very good. Both predictors have a very high value of ROC curve integral (0.954 in both cases). This ensures that the predictors can be finely tuned (by introducing a bias parameter in the prediction, i.e. classification phase) to increase sensitivity (or specificity), while retaining a fairly high level of specificity (or sensitivity in the reverse case).

TABLE II
POSITION-SPECIFIC AMINO ACID PROPENSITIES NEAR C-ENDS OF A-HELICES; C1 IS THE FIRST RESIDUE INSIDE HELIX, FOLLOWED BY C2, C3, ETC.; CCAP IS THE FIRST RESIDUE OUTSIDE HELIX, FOLLOWED BY C',C? AND C?'.

	Inside helix						Outside helix			
	C6	C5	C4	C3	C2	C1	Ccap	C'	C?	C?'
A	**1.51**	1.43	**1.66**	**1.69**	1.37	1.49	1.13	0.64	0.81	0.82
C	0.70	0.71	1.09	1.01	0.71	0.78	0.98	0.65	0.72	0.77
D	0.80	0.69	0.54	0.63	0.61	0.62	0.80	0.98	1.33	1.25
E	1.29	1.23	1.11	1.21	1.49	1.33	0.80	0.79	1.02	0.98
F	0.93	1.18	1.29	1.04	0.78	1.05	0.81	0.62	0.92	0.88
G	0.35	0.28	0.34	0.31	0.23	0.29	**2.41**	**2.55**	1.17	0.94
H	0.90	0.83	0.82	0.83	1.03	1.24	1.42	0.96	1.14	1.01
I	1.17	1.44	1.31	1.01	1.16	0.71	0.40	0.62	0.90	0.91
K	1.18	1.14	1.07	1.43	**1.77**	1.40	1.19	1.23	1.23	1.14
L	1.38	1.49	**1.56**	**1.70**	1.47	**1.50**	0.95	0.68	0.92	0.86
M	1.32	1.34	**1.66**	**1.56**	1.28	1.25	0.93	0.64	0.80	0.72
N	0.75	0.68	0.58	0.71	0.70	1.05	**1.70**	1.23	1.06	1.05
P	0.15	0.08	0.10	0.10	0.02	0.00	0.00	**2.08**	**1.59**	**1.94**
Q	1.40	1.24	1.19	1.17	1.37	1.29	1.25	0.98	0.90	0.84
R	1.33	1.30	1.15	1.45	1.50	1.27	1.07	1.00	0.93	0.93
S	0.66	0.62	0.62	0.72	0.71	1.10	1.11	0.96	0.91	0.98
T	0.75	0.67	0.59	0.52	0.71	1.00	0.66	0.77	0.85	1.05
V	1.09	1.02	1.00	0.67	0.87	0.57	0.43	0.59	0.78	0.99
W	0.95	1.28	1.36	1.08	0.88	0.61	0.46	0.54	0.68	0.74
Y	0.89	1.19	1.18	0.92	0.82	1.12	0.91	0.57	1.01	0.88

TABLE III
EVALUATION OF PREDICTORS OF A-HELIX AND H, E & C SEGMENT ENDS

	True Accuracy (%)	Sensitivity (%)	Specificity (%)	ROC Curve Integral
PNA	88.2 ± 0.6	87.5 ± 1.0	88.9 ± 0.8	0.954 ± 0.005
PCA	88.6 ± 0.7	89.2 ± 1.4	88.0 ± 0.6	0.954 ± 0.005
PNH	80.5 ± 0.7	74.9 ± 1.2	86.1 ± 0.7	0.890 ± 0.006
PCH	78.2 ± 0.7	74.2 ± 0.9	82.2 ± 1.0	0.865 ± 0.005
PNE	77.3 ± 0.6	74.4 ± 0.8	80.2 ± 0.9	0.855 ± 0.006
PCE	76.7 ± 0.6	75.1 ± 0.7	78.3 ± 1.1	0.849 ± 0.006
PNC	74.7 ± 0.5	77.1 ± 0.6	72.2 ± 0.8	0.826 ± 0.005
PCC	75.5 ± 0.6	76.8 ± 0.7	74.1 ± 0.8	0.839 ± 0.006

Performance of the other six predictors was also quite good. Out of 3 segment types, predictors of the ends of H segments (PNH, PCH) achieved the best accuracy, though they are not as accurate as predictors of ends of a-helices. The reason for this difference is that 3_{10}-helix and p-helix residues, which are often surrounding a-helices, introduce noise that hinders prediction of the ends of H segments. Ends of C segments are least predictable, which is an expected result when we consider that C segments have the least regular structure.

Prediction of secondary structure using the output of H, E and C segment-ends predictors

We first compare the effect of various window sizes on the quality of prediction. The values of performance indicators gradually increased as window size increases, and they practically converged when window size was 15 (Table IV). It is very important that the improvement of Q_3 results from the improvement of two component accuracies, Q_3H and Q_3E, while Q_3C stays practically the same (i.e. none of the component accuracies decrease). In further experiments, we used a window of 15 residues. This window size was previously estimated to be optimal for predictors based on sequence profile data [3,21,22].

We proceed with ranking input attributes – obtained from different segment-ends predictors – by importance for quality of prediction. Table V contains results of predictors that use only input attributes obtained from 2 out of 3 pairs of the segment-ends predictors (one of the pairs is omitted in each turn), as well as the predictors that use all input attributes. Results show that omitting input attributes from one pair of segment-ends predictors does not have a great impact on prediction accuracy, therefore indicating that there is some redundancy in the data that makes it less noise-sensitive.

Improving basic predictors of secondary structure by introducing the output of segment-ends predictors

Results for our basic secondary structure predictor based on sequence profile data (PSS_{PROF}), and for secondary structure predictor based on the output of segment-ends predictors (PSS_{HEC}) is listed in Table VI. PSS_{HEC} is not only comparable to PSS_{PROF}, it actually has a small advantage in both Q3 (0.3%) and SOV3 (1.0), though the standard deviation intervals still overlap. This is a very important result since PSS_{HEC} uses only 6 input attributes per residue position in a window (compared with 20+1 for PSS_{PROF}), and values for those 6 input attributes come from segment-ends predictors that are far from perfect (note that segment-ends predictors use the same type of input as PSS_{PROF}.) In other words, a window of 15*6=90 segment-ends predictions appears to carry as much information as a window of 15*21=315 original attributes.

When comparing single predictors – components of ensemble predictors PSS_{PROF} and PSS_{HEC} – we noted an interesting phenomenon. Difference in Q_3 accuracy of ensemble and single predictors is only +0.3% for PSS_{HEC}, compared to +2.0% for PSS_{PROF}. Even more interesting is the difference in SOV_3 between ensemble and single predictors: +2.0 for PSS_{HEC}, compared to +7.0 for PSS_{PROF}. We believe that the reason for this is that PSS_{HEC} is less subject to the "curse of dimensionality" problems. Component predictors are trained on smaller training sets (only a sixteenth part of the whole dataset). PSS_{HEC} takes 90 inputs and small training sets appear to be sufficient to successfully train component predictors. This is the reason why there is only a small improvement when these component predictors are integrated into an ensemble. On the other hand, PSS_{PROF} takes 315 inputs. Small training sets appear to be insufficient for training component predictors. When these (far from optimal) component predictors are integrated into an ensemble, the accuracy of prediction increases to a much higher level.

TABLE IV
EVALUATION OF SECONDARY STRUCTURE PREDICTORS THAT USE INPUTS OBTAINED FROM SEGMENT-ENDS PREDICTORS (COMPARISON OF VARIOUS INPUT WINDOW SIZES); Q_3 IS PER-RESIDUE ACCURACY; Q_3H, Q_3E, Q_3C ARE PER-RESIDUE ACCURACIES ON H, E AND C PARTS OF PROTEINS; SOV_3 IS SEGMENT OVERLAP MEASURE

	Q_3 (%)	Q_3H (%)	Q_3E (%)	Q_3C (%)	SOV_3
$PSS_{HEC;7}$	77.7 ± 0.5	79.7 ± 0.7	66.4 ± 0.9	82.5 ± 0.6	73.8 ± 0.7
$PSS_{HEC;9}$	77.9 ± 0.5	80.0 ± 0.7	67.0 ± 0.8	82.4 ± 0.6	74.4 ± 0.7
$PSS_{HEC;11}$	78.1 ± 0.5	80.1 ± 0.7	67.5 ± 0.8	82.4 ± 0.6	74.7 ± 0.7
$PSS_{HEC;13}$	78.1 ± 0.5	80.2 ± 0.7	67.7 ± 0.8	82.3 ± 0.6	74.8 ± 0.6
$PSS_{HEC;15}$	78.1 ± 0.5	80.2 ± 0.7	67.9 ± 0.7	82.2 ± 0.6	74.9 ± 0.7

TABLE V
COMPARISON OF PREDICTORS OF SEC. STRUCTURE THAT USE INPUTS OBTAINED FROM VARIOUS SELECTIONS OF SEGMENT-ENDS PREDICTORS (WINDOW SIZE 15)

	Q_3 (%)	Q_3H (%)	Q_3E (%)	Q_3C (%)	SOV_3
PSS_{HE}	77.7 ± 0.4	80.1 ± 0.7	67.5 ± 0.8	81.5 ± 0.5	74.5 ± 0.7
PSS_{EC}	77.9 ± 0.5	79.9 ± 0.7	67.2 ± 0.7	82.2 ± 0.5	74.5 ± 0.7
PSS_{HC}	78.0 ± 0.5	80.0 ± 0.6	67.3 ± 0.7	82.3 ± 0.6	74.7 ± 0.7
PSS_{HEC}	78.1 ± 0.5	80.2 ± 0.7	67.9 ± 0.7	82.2 ± 0.6	74.9 ± 0.7

TABLE VI
EVALUATION OF STAGE-ONE PREDICTORS (WINDOW SIZE 15)

	Q_3 (%)	Q_3H (%)	Q_3E (%)	Q_3C (%)	SOV_3
PSS_{PROF}	77.8 ± 0.4	80.3 ± 0.6	67.8 ± 0.7	81.3 ± 0.6	73.9 ± 0.7
PSS_{HEC}	78.1 ± 0.5	80.2 ± 0.7	67.9 ± 0.7	82.2 ± 0.6	74.9 ± 0.7
$PSS_{PROF+HEC}$	78.3 ± 0.5	80.9 ± 0.6	69.0 ± 0.7	81.4 ± 0.6	74.5 ± 0.7
PSS_{MEAN}	78.6 ± 0.4	80.6 ± 0.6	68.7 ± 0.7	82.4 ± 0.5	75.3 ± 0.7

We continue by evaluating two predictors that use both sequence profile data and segment-ends predictors data. $PSS_{PROF+HEC}$ is a neural network based predictor that takes both types of inputs, 415 in total (Table VI). It achieves an even higher Q_3 accuracy than PSS_{HEC}, but its SOV_3 score is lower (although it is still higher than SOV_3 score for PSS_{PROF}). We noted that the average SOV_3 score of its component predictors was quite low (67.9), as was the case with PSS_{PROF}.

We believe that this is due to the imbalance in the number of inputs of two types (315 versus 90), and that sequence-profile based inputs dominate the training and prediction process.

PSS_{MEAN} also uses both types of inputs, but does not "mix" them. It separately obtains raw outputs (triplets of real numbers) from both PSS_{PROF} and PSS_{HEC} and then produces its own prediction based on mean values of two row outputs. It achieved the best Q_3 accuracy (78.6%) and the best SOV_3 score (75.3) of all first-level predictors. These results are significantly better than those of basic PSS_{PROF}, because their standard deviation intervals do not overlap.

Stage-two predictors

Results for various stage-two predictors are listed in Table VII. Basic stage-two predictor S2[PSS_{PROF}] greatly improved the Q_3 accuracy and SOV_3 score of its basic stage-one predecessor PSS_{PROF}. Stage-two predictor S2[PSS_{HEC}] did not provide such a great improvement to results of its stage-one predecessor. We believe that this model for stage-two prediction is inappropriate for the output produced by PSS_{HEC}. S2[PSS_{PROF}] probably outperformed S2[PSS_{HEC}] because PSS_{PROF} produces output with higher level of noise than PSS_{HEC}, so S2[PSS_{PROF}] is able to generalize better when trained on such noisy input.

Other stage-two predictors that involve data obtained from predictors of segment-ends (S2[$PSS_{PROF+HEC}$] and S2[PSS_{PROF}, HEC]) performed better than basic predictor S2[PSS_{PROF}]. However, these differences are smaller than standard deviations. Additionally, S2[PSS_{PROF}, HEC] has a higher Q_3 accuracy, but a lower SOV_3.

Predictor $S2_{MEAN}$ seems to inherit qualities of both its components: high Q_3 accuracy of S2[PSS_{PROF}, HEC], and high SOV_3 score of S2[$PSS_{PROF+HEC}$]. It achieved the best Q_3 accuracy (79.2%) and the best SOV_3 score (76.4) of all evaluated predictors.

TABLE VII
EVALUATION OF STAGE-TWO PREDICTORS

	Q_3 (%)	Q_3H (%)	Q_3E (%)	Q_3C (%)	SOV_3
S2[PSS_{PROF}]	78.6 ± 0.5	80.6 ± 0.7	68.7 ± 0.7	82.6 ± 0.6	76.1 ± 0.8
S2[PSS_{HEC}]	78.4 ± 0.5	80.6 ± 0.7	67.7 ± 0.7	82.6 ± 0.6	75.4 ± 0.7
S2[$PSS_{PROF+HEC}$]	78.8 ± 0.5	80.9 ± 0.7	69.0 ± 0.7	82.6 ± 0.6	76.4 ± 0.7
S2[PSS_{PROF},HEC]	79.3 ± 0.5	81.5 ± 0.7	69.6 ± 0.7	82.7 ± 0.6	75.7 ± 0.8
$S2_{MEAN}$	79.2 ± 0.5	81.3 ± 0.7	69.5 ± 0.7	82.9 ± 0.6	76.4 ± 0.7

IV. CONCLUSION

Position-specific amino acid propensities for positions near the ends of a-helices, listed in Tables I and II are consistent with previous findings and suggest that it is possible to predict the ends of a-helices (problems N_A and C_A). This assumption was clearly confirmed by our experiments.

Results of the next set of experiments show that it is feasible to generalize problems N_A and C_A to other types of secondary structure segments (H, E and C).

We used outputs from H, E and C segment-ends predictors as input for secondary structure prediction. Secondary structure predictors that use only data obtained from segment-ends predictors have performance comparable to predictors that use larger-dimensional sequence profile-based data.

Success of prediction of secondary structure only from the outputs of segment-ends predictors brings up a challenge to find the best way of improving regular secondary structure predictors by incorporating data obtained from segment-ends predictors.

We tackled this challenge by using two approaches. A predictor that uses both types of input simultaneously shows some improvement over a basic predictor, but it is not better than a predictor that uses only data from segment-ends predictors. A much better approach was to obtain two separate "raw" outputs (one from the basic predictor using data from the sequence profile, another from the predictor that uses only output from segment-ends predictors) and perform prediction based on the means of these two "raw" outputs. This predictor has significant improvement compared to the basic predictor, +0.8% for Q_3 and +1.4 for SOV_3 score.

We also tried to improve two-stage prediction by incorporating data obtained from segment-ends predictors. It is not yet clear how this can be done in an efficient way. We were able to achieve some improvement, although it was not statistically significant. We believe that the simple method that we used for all stage-two predictors might not be appropriate for filtering of outputs from stage-one predictors that already produce very accurate prediction.

Protein secondary-structure prediction is a mature problem, and there are some indications that methods might be reaching a theoretical upper-limit of Q_3 accuracy. This makes the improvement of Q_3 accuracy and particularly SOV_3 score that we achieved for basic stage-one predictor even more significant. Open questions remains whether prediction of segment-ends can be used to improve current state-of-the-art secondary-structure predictors, and whether this could lead to improvement of methods that use secondary-structure prediction as intermediate source of data.

ACKNOWLEDGEMENT

This work was supported by NIH grant R01 LM007688-01A1 to A.K. Dunker and Z. Obradovic.

REFERENCES

[1] C.A. Orengo, J.E. Bray, T. Hubbard, L. LoConte, and I. Sillitoe, "Analysis and assessment of ab initio three-dimensional prediction, secondary structure, and contacts prediction," Proteins. 1999; Suppl 3: 149–170.

[2] L.A. Kelley, R.M. MacCallum, and M.J. Sternberg, "Enhanced genome annotation using structural profiles in the program 3D-PSSM," J Mol Biol. 2000 June 2; 299(2): 499–520.
[3] J.A. Cuff and G.J. Barton, "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction," Proteins. 1999 March 1; 34(4): 508–519.
[4] B. Rost and C. Sander, “Prediction of protein secondary structure at better than 70% accuracy,” J. Mol. Biol. 1993; 232: 584-599.
[5] D.T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," J Mol Biol. 1999 September 17; 292(2): 195–202.
[6] D. Przybylski and B. Rost, "Alignments grow, secondary structure prediction improves," Proteins. 2002 February 1; 46(2): 197–205.
[7] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles," Proteins. 2002 May 1; 47(2): 228–235.
[8] B. Rost, C. Sander, and R. Schneider, "Redefining the goals of protein secondary structure prediction," J Mol Biol. 1994 January 7; 235(1): 13–26.
[9] A. Zemla, C. Venclovas, K. Fidelis, and B. Rost, "A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment," Proteins. 1999 February 1; 34(2): 220–223.
[10] J.S. Richardson and D.C. Richardson, "Amino acid preferences for specific locations at the ends of alpha helices," Science, 1988 June 17; 240(4859): 1648–1652.
[11] L.G. Presta and G.D. Rose, "Helix signals in proteins," Science, 1988 June 17; 240(4859): 1632–1641.
[12] R. Aurora and G.D. Rose, "Helix capping," Protein Sci. 1998 January; 7(1): 21–38.
[13] D.E. Engel, W.F. DeGrado, “Amino acid propensities are position-dependent throughout the length of alpha-helices,” J Mol Biol. 2004 Apr 9;337(5):1195-205.
[14] P. Radivojac, Z. Obradovic, C.J. Brown, A.K. Dunker, “Prediction of boundaries between intrinsically ordered and disordered protein regions,” Pac Symp Biocomput. 2003;:216-27
[15] http://swift.cmbi.kun.nl/whatif/select/
[16] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," Nucleic Acids Res. 1997 September 1; 25(17): 3389–3402
[17] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," Biopolymers. 1983 December; 22(12): 2577–2637.
[18] http://cubic.bioc.columbia.edu/eva/
[19] P. Werbos, “Beyond regression: New tools for prediction and analysis in the behavioral sciences,” Ph.D. Thesis, Harvard Universiry, 1974.
[20] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: The RPROP algorithm,” Proc. Int. Conf. Neural Networks, San Francisco, 586–591, 1993.
[21] L.H. Wang, J. Liu, Y.F. Li and H.B. Zhou, “Predicting protein secondary structure by a support vector machine based on a new coding scheme,” Genome Inform Ser Workshop Genome Inform. 2004;15(2):181-90.
[22] N. Qian and T.J. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," J Mol Biol. 1988 August 20; 202(4): 865–884.