BIOINFORMATICS

Vol. 17 no. 6 2001
Pages 551–556

Prediction of quaternary structure from primary structure

Robert Garian

School of Computational Sciences, George Mason University, USA

Received on May 8, 2000; revised on February 5, 2001; accepted on February 15, 2001

ABSTRACT

Motivation: The 'sequence implies conformation' principle has been the motivation for the construction of numerous systems of secondary and tertiary structure prediction. Computational experiments have shown that this principle can now be extended to quaternary structure prediction. This work appears to be the first effort to predict quaternary structure properties from sequence.

Results: The software developed to conduct these experiments was the Quaternary Structure Explorer (QSE). Successful rule-based classifiers have been found that can discriminate between the primary sequences of homodimers and non-homodimers.

Availability: The homodimer classifier can be accessed at http://www.mericity.com

Contact: rgar@science.gmu.edu

INTRODUCTION

In a classic experiment carried out in the 1950s, denatured ribonuclease was found to be fully restored in its catalytic activity after the denaturing agent was removed (Anfinsen *et al.*, 1961). This experiment provided strong evidence for what is now a generally accepted principle: the amino acid sequence of most, if not all, proteins contains all the information needed to fold the protein into its correct three-dimensional structure. At the next level of protein organization, tertiary structures associate into quaternary structures forming multimeric proteins. The association of tertiary structure subunits depends upon the existence of complementary 'patches' on their surfaces. The patches are buried in the interfaces formed by the subunits and, thus, play a role in both tertiary and quaternary structure. This suggests that primary sequences contain quaternary structure information.

Klotz *et al.* (1975) reviewed the nature of the quaternary structure of proteins in a now-classic paper. They distinguished a number of quaternary structure properties such as stoichiometric constitution, the geometric arrangements of the subunits, the assembly energetics, intersubunit communication, and their functional aspects. In the present work, only the *number* of subunits (distinct chains) in a homo-oligomeric protein was considered; this aspect of protein stoichiometry will be referred to as the *mericity* of the assembled protein.

The actual quaternary properties of proteins must be determined empirically. The goal of the present work, however, was to determine the presence or absence of mericity information in the primary sequences of proteins using a machine learning method. A software system was designed to prepare data and to generate classifiers for the primary sequences of proteins of known quaternary structure. A classification system capable of distinguishing among the primary sequences of different mericities would allow the rejection of the hypothesis that protein sequences do not contain any quaternary structure information.

SYSTEM AND METHODS

Machine learning is an active area of research in artificial intelligence that investigates methods for making and evaluating classification decisions and predictions by discovering the patterns that exist in data. It is based on the idea that objects can be described by the values of their attributes. If an object can be described with n attributes, then an n-tuple of values of its attributes, an *example*, can be used to represent it. Let X be a sample of N examples, and let $x_i = (a_1, a_2, \ldots, a_n)$ be an example in X, where a_j is the value of the jth attribute of x_i. Assume that each example has somehow been assigned its correct label y_i. There is an unknown function f such that $f(x_i) = y_i$. A *learning method* attempts to construct a function h that approximates f (Dietterich, 1997).

The quality of h as an approximation can be measured by its misclassifications of examples. If the examples of X are classified into two classes labeled 'positive' or 'negative', then each application of h to an example in X can result in only one of four types of classification: a True Positive (TP), a False Negative (FN), a False Positive (FP), or a True Negative (TN). The corresponding four classification frequencies constitute the *confusion matrix* of h. There is also a related *cost matrix* that assigns relative weights to the frequencies; we will assume, however, that the costs are all equal.

The *apparent error rate* of h is defined as $(\mathrm{FP} + \mathrm{FN})/\mathrm{N}$.

It measures the performance of classifier h on the members of X. While a low apparent error rate is desirable, it is not a sufficient criterion for accepting a classifier. A good h function is expected to perform well on new instances of similar examples that are not in X; h's domain should generalize to examples beyond the immediate training data. In those cases when h fails to form a generalization and yet has a low apparent error rate, it is said to have overfitted the data.

A basic requirement for good classifier performance is that the unseen examples be drawn from the same population as the original sample used in its construction (Weiss and Kulikowski, 1990; Quinlan, 2000). Each sample has an associated distribution that will be reflected in the training data used to construct the classifier. A classifier's accuracy is always measured with respect to the distribution of its training data (Quinlan, 2000).

Classifier performance can be improved by using more training examples. As the number of randomly drawn examples from the population increases without limit, the apparent error rate will approach the *true error rate*. In a practical machine learning application, however, the sample size may be small and the true error rate can only be estimated.

The k-fold cross-validation method is a robust way of estimating the true error rates of classifiers for samples containing at least 100 examples (Weiss and Kulikowski, 1990). This re-sampling method randomly partitions the training set into k blocks of examples of approximately equal size. A classifier is then trained on $k-1$ blocks. The remaining block is set aside as a test block. This process is repeated for k iterations, each time setting aside a different test block. Thus, each of the blocks will serve both as part of the training set and as a test block.

The average of the error rates observed for each of the k classifiers is the cross-validated error rate. It is known to provide an honest estimate of the true error rate through direct simulation of the performance of classifiers on unseen examples. Rost and Sander (1995) have discussed the value of cross-validation in protein structure prediction.

The Quaternary Structure Explorer (QSE) system was developed to test the mericity hypothesis using the C4.5 machine-learning program (Quinlan, 1993). This program generates decision-tree and rule-based models of data. A decision tree is a labeled graph (similar to a programmer's flowchart) consisting of *decision nodes*, which specify tests of attribute values, and *leaf nodes*, which specify classes. Let T be a set of training examples, then a C4.5 decision tree is constructed as follows: if T is empty, assign the label of the most frequent class to T. If T is not empty and contains only examples of the same class, then create a leaf node labeled with that class. If T contains a mixture of examples from different classes, then create a decision node and partition T into disjoint subsets based on some predefined splitting criteria, so that each subset corresponds to a possible outcome. Extend a branch from the decision node to each block of the partition. Recursively apply these rules to each of the blocks generated, so that they become more homogeneous with each successive test. Stop when each block contains examples of only a single class.

The final tree consists of a root node and a set of paths that follow the branches and terminate at the leaves. A path in a decision tree can be translated into a conjunction of tests that specify a leaf. After some simplifications, the set of all such conjunction–leaf pairs becomes a rule-based model of the training data. A rule-based model can also be viewed as an approximation function h that maps examples to leaves.

Decision-tree models have been shown to be relatively insensitive to *imbalanced* training sets (Drummond and Holte, 2000). They are known to perform well even though the distribution may be skewed by having many examples of one class and few of another. The training set used by QSE in these experiments, however, was only slightly skewed toward the homodimers. The various remedies proposed to alleviate imbalance in training sets and other learning methods have themselves raised many theoretical issues, some of which remain unsettled (Provost and Fawcett, 1997; Drummond and Holte, 2000).

ALGORITHM

The QSE program requires a set of parameters $P = \{S, A, W, D, F, k_1, k_2, p\}$, where S is the target set of protein sequences, A is a set of amino acid indices, W is a set of window sizes used for smoothing, D is a set of numbers specifying the number of features to be used in a feature vector (n-tuple) representation of a sequence, F is a set of feature extraction functions, k_1 is the number of folds to be used in k_1-fold cross validation, k_2 specifies the number of trials (resampling runs), and p is the proportion of the target sequences to be used in the training set with the remainder held back for testing. Pseudocode for the QSE system is shown in Figure 1. The code shows that QSE loops over each of the experimental factors one at a time to create an experiment E.

Sequence data

In the present experiment, S was a set of homoligomeric sequences (the target set) obtained from Release 34 of the SWISS-PROT database (Bairoch and Apweiler, 1996). It was limited to the prokaryotic, cytosolic subset of homo-oligomers in the database in order to eliminate membrane proteins and other specialized proteins. The database consisted of 1639 homo-oligomeric protein sequences, 914 of which were homodimers and 725 non-homodimers.

```
Given parameters P={S,A,W,D,k1,k2,F,p}
Generate Experiment E:
  For each vector dimension
    For each function
      For each window size
        For each scale
          For each sequence in target set S
            Generate feature vector
Repeat k2 times
  For each subexperiment in E
    Randomly partition the subexperiment's feature vectors into a training set, T,
    containing p|S| examples and a holdout set, H, containing the
    remaining examples
      Perform k1-fold cross-validation with C4.5 on the examples
        For each of the classifiers
          Convert decision-tree classification model to a rule-based model
          Extract the confusion matrix
          Convert rule-based model to an executable program
          Apply the executable to H
          Save all error data
          Save executable
  Compute performance measures from error file
  Rank classifiers based on measures
```

Fig. 1. Pseudocode for the QSE system.

Amino acid indices

The set, A, of 401 amino acid indices was obtained from the April 1996 version of the AAindex database (Kidera *et al.*, 1985), which was first developed in 1985 by Kidera *et al.* An amino acid index is a list of 20 numerical values corresponding to physical, chemical, and biochemical properties of the 20 common amino acids. These indices were found to cluster into a small number of groups: α and turn propensity, β propensity, composition, hydrophobicity, physicochemical properties, and other properties (Nakai *et al.*, Tomii and Kanehisa, 1996). Recently, the entire set of 402 indices has been summarized by just two indices (Hagerty *et al.*, 1999).

Other parameters

In this experiment, only 401 of the indices were used because one index was incomplete. The remaining parameter settings were as follows: $D = \{5, 10\}$, $W = \{1, 29\}$, $k_1 = 10$, $k_2 = 5$, and $P = 0.66$. Two feature extraction functions were specified in F. Using these parameters, by varying one factor at a time, QSE generated $401 \times 2 \times 2 \times 2 = 3208$ distinct sets of training data or *subexperiments*.

Amino acid profiles and feature extraction

A QSE subexperiment consists of a set of feature vectors (examples) derived from the set of target sequences. To obtain the feature vectors, the sequences are first transformed into protein sequence profiles using an amino acid index as a substitution table to create a discrete sequence of values, the *profile*. The profiles are smoothed by sliding a window of specified length along the sequence and calculating the mean of the values within each window to form a new profile. A feature extraction function is then applied to the smoothed profile to produce a d-dimensional feature vector.

QSEs feature extraction functions were designed to extract local compositional characteristics as well as differences in the relative intensities of the properties represented by the profile's amino acid index. The profile is partitioned into intervals along the ordinate. The number of points of the profile within an interval is then counted. This simple binning function was designated internally by Cumulative Density Distribution Vector (CDDV). A variant of this function was also defined in which the bins represent sums rather than counts of profile values.

These functions were designed after consideration of some of the results of recent work on protein–protein interfaces by a number of researchers. Their work has shown that interfaces have complex structures and compositions. In addition, the hydrophobic cores of interfaces are variable and not well-conserved.

In a review of protein dimer structures, Jones and Thornton (1995) characterized the protein–protein interactions at interfaces. These interactions are driven by both the hydrophobic effect (Klotz *et al.*, 1975) and the specificity of the subunits that is due to the complementarity of their 'surface patches'. Shape complementarity is the correspondence of projecting regions of one patch with depressed regions of the other. It was found that the interfaces between dimers are more hydrophobic than the exterior, but less hydrophobic than the interior of a subunit and that 'interfaces are discontinuous, segmented surfaces with between 2 and 15 segments and a mean of 5.5'. In related work on protein–protein interfaces Tsai *et al.* (1997) analyzed the data for 362 representative oligomeric interfaces and found that the amino acid composition of the interfaces was more similar to the composition of the overall protein than to the surface of the protein.

In a survey of the morphology of 136 homodimeric interfaces in the Protein Data Bank, Larsen *et al.* (1998) found that the pattern of hydrophilicity was quite variable. A third of the interfaces had a recognizable hydrophobic core (a hydrophobic patch surrounded by a ring of polar interactions). The remaining two-thirds of the proteins had a mixture of hydrophobic patched, polar interactions, and water molecules scattered over the interface area. Some of the proteins are associated by extensive interdigitation of their subunit chains (Larsen *et al.*, 1998). Grishin and Phillips (1994) addressed the question of the conservation of residues in subunit interfaces and found that 'amino acid residues that make up the hydrophobic core are not well conserved and evolve nearly as rapidly as the overall protein sequence, despite their importance to the integrity of the protein structure and function'.

These results on the nature of protein–protein interfaces

constrained the possibilities for the design of a quaternary feature extraction function. They effectively eliminated the sequence alignment approach from consideration and suggested the use of the amino-acid profile approach. The use of profiles also suggested the application of simple signal processing methods to protein sequences.

Model building

Once the set of profiles are generated and the feature vectors are created, QSE randomly partitions them into a training set, T, and a *holdout* set, H. The holdout set is used for testing rule-based classifier models constructed by 10-fold cross-validation with C4.5. The confusion matrix data for each classifier are collected and later used to rank the classifiers. In the experiment conducted with the 401 amino acid indices from AAindex, this process was repeated five times and generated a total of $3208 \times 10 \times 5 = 160\,400$ classifiers. The classifiers were applied to the data in the holdout sets created during each of the five runs of the experiment to obtain their apparent error rates.

Performance measures

QSE uses several performance measures to evaluate classifiers. All are derived from the confusion matrix values for a given classifier, which consists of the frequencies TP, TN, FP, and FN. The overall accuracy of a classifier is defined as $1 - (\text{apparent error rate}) = (\text{TP} + \text{TN})/N$, where N is the total number of examples. The TP rate, TPR, is defined as $\text{TP}/(\text{TP} + \text{FN})$ and the FP rate, FPR, is defined as $\text{FP}/(\text{FP} + \text{TN})$. Since TPR and FPR are independent of N, they are better predictors than those that vary with changes in N.

QSE plots the (FPR, TPR) points for all the classifiers in what is known as ROC-space (Provost and Fawcett, 1997). This is a region in the x, y-coordinate plane defined by the corners of the unit square: (0, 0), (1, 0), (1, 1), (0, 1). The diagonal from (0, 0) to (1, 1) is called the *chance line*. The point (0, 1) corresponds to the perfect classifier. Those points closer to the perfect classifier in ROC-space are the better classifiers, and those on the chance line are no better than random guessing. Given two points in ROC-space, the point higher and to the left of the other represents the better classifier.

Results

A considerable number of the classifiers were found to perform at a level better than random guessing. Thus, this QSE experiment was successful in finding parameters that can be used to construct classifiers capable of discriminating between homodimers and non-homodimers. The top classifier found during this experiment had an apparent error rate of 34% and an error rate of 28% on the holdout set. Figure 2 is a QSE ROC-space plot showing the variation in the overall accuracy of 160 400 classifiers generated during the experiment. A line connecting the points with the greatest ordinates for each FPR value represents the convex hull of the cloud of points. This line of points can be used to choose the best classifiers with a given FPR.

Table 1. Results of a 10-fold cross-validation run using the parameters found by QSE

	Decision tree		Rules	
	Size	Errors (%)	Number	Errors (%)
Mean	111.3	31.1	58.0	30.0
Standard error	3.5	0.8	3.1	1.0

The highest scoring combination of parameters associated with this classifier was $D = \{10\}$, $W = \{1\}$, $F = \{\text{CDDV}\}$ and $A = \{\text{index NAKH900111 (Nakashima } \textit{et al.}\text{, 1990)}\}$. These parameters were used in a subsequent focused experiment using C5, a later version of C4.5, to investigate the true error rate of the classifiers associated with these parameters. The mean error rate obtained with a 10-fold cross-validation run was 30% for the rule-based model. This is an estimate of the true error rate for classifiers generated with these parameters.

Another aspect of classifier performance is the *complexity of fit*. This term refers to the preference for classifiers having fewer decision nodes and fewer rules. Since they have succeeded in forming more concise generalizations, they are less likely to overfit the data. The average decision-tree model required 111 nodes and the average rule-based model required 58 rules as shown in Table 1.

There are many possible performance measures that can be computed for a binary classifier. Many of these have their origin in medical statistics, but they are also used in information-retrieval and signal-detection theory. The Matthews correlation coefficient, which varies from -1 to 1, has its origin in early work on secondary structure prediction (Matthews, 1975). A value of zero corresponds to random predictions (the chance line in ROC-space). It provides a single value with which to compare classifiers at the expense of some loss of information. Table 2 shows the values of some of the common performance measures that can be derived from the frequencies provided by the confusion matrix of the homodimer classifier.

DISCUSSION

The results of computational experiments with QSE support the alternative to the mericity hypothesis: the feature vectors used to generate the decision-tree and rule-based models have been shown to contain mericity information, hence the primary sequences of homo-oligomeric proteins contain quaternary information. The feature

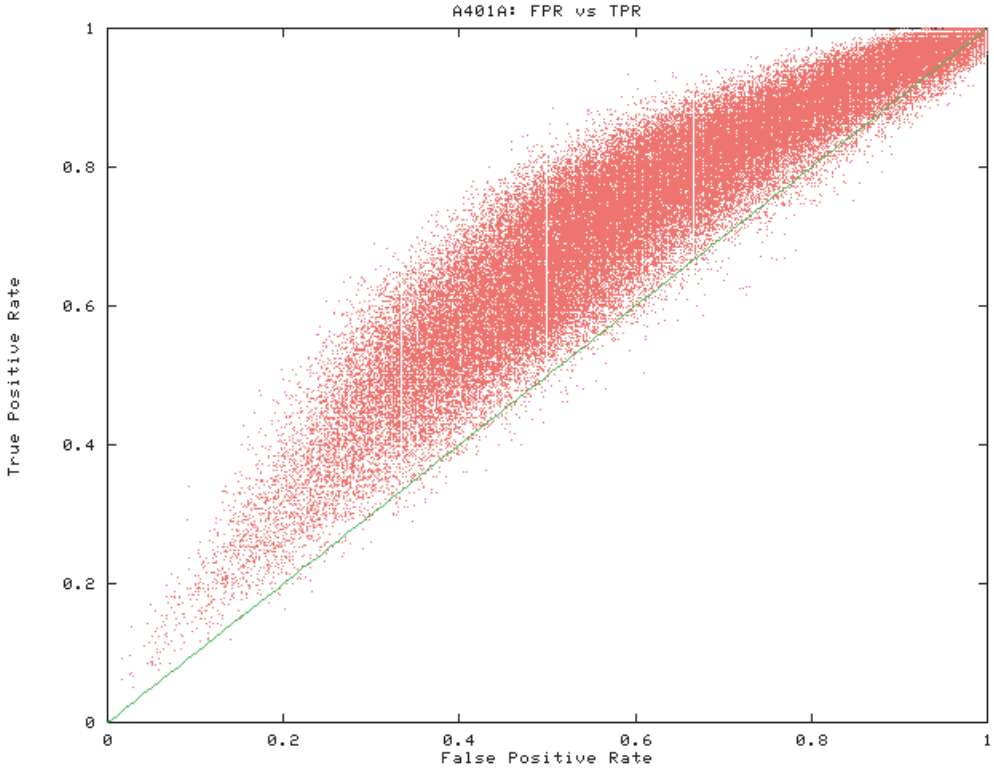

Fig. 2. ROC-space plot of the (FPR, TPR) pairs of all classifiers generated in the experiment.

Table 2. Some common performance measures derived from the confusion matrix

TN	FN	FP	TP
433	200	292	714

Sensitivity	0.781
Specificity	0.597
Positive predictive value	0.709
Negative predictive value	0.684
Apparent accuracy	0.699
TP rate	0.781
FP rate	0.402
Matthews correlation coefficient	0.386

vectors appear to capture essential information about the composition and hydrophobicity of the residues in the surface patches that are buried in the interfaces of associated subunits. This information is provided in the distribution of profile values associated with a suitable amino acid index.

QSE experiments perform an empirically focused search for quaternary structure-revealing indices in parameter space. The use of these indices significantly constrains the parameter space to structurally meaningful regions. If the AAindex database did not exist, it would have been necessary to search the entirety of the parameter space.

These computational experiments have shown the usefulness of the sequence profile approach to protein classification. It should be considered as an alternative when the sequence alignment method fails or is inappropriate. Protein sequence profiles were found to provide a natural way of mining the SWISS-PROT and AAindex databases for information about quaternary structure.

CONCLUSION

Classifiers were found that are capable of discriminating between homodimers and non-homodimers. The best of these, at the present time, has an estimated true error rate of 30%. Predictions are, of course, probabilistic; only experimental work can ultimately establish the quaternary properties of a particular protein sequence. Areas of future work with quaternary classifiers that are currently being investigated are the construction of classifiers for the higher homo-oligomers and the inference of interface residues from sequence.

ACKNOWLEDGEMENTS

The author is grateful to Evgeni Selkov for suggesting the general direction of this work and to his dissertation committee for their support. The insightful comments of the anonymous reviewers are also greatly appreciated.

REFERENCES

Anfinsen,C.B., Haber,E., Sela,M. and White Jr.,F.H. (1961) The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proc. Natl Acad. Sci. USA*, **47**, 1309–1314.

Bairoch,A. and Apweiler,R. (1996) The SWISS-PROT protein data bank and its new supplement TrEMBL. *Nucleic Acids Res.*, **24**, 21–25.

Dietterich,T.G. (1997) Machine learning research: four current directions. *AI Magazine*, **18**, 97–136.

Drummond,C. and Holte,R.C. (2000) Exploiting the cost (in) sensitivity of decision tree splitting criteria. In *Proceedings of the 17th International Conference on Machine Learning (ICML'2000)*, pp. 239–246.

Garian,R. (2000) *Prediction of dimeric structure of proteins from amino acid sequences*, PhD Dissertation, School of Computational Sciences, George Mason University.

Goldberg,M. (1985) The second translation of the genetic message: protein folding and assembly. *TIBS*, 388–391.

Grishin,N. and Phillips,M.A. (1994) The subunit interfaces of oligomeric enzymes are conserved to a similar extent to the overall protein sequences. *Protein Sci.*, **3**, 2455–2458.

Hagerty,C.G., Muchnik,I., Kulikowski,C. and Kim,S.-H. (1999) Two indices can approximate four hundred and two amino acid properties. In *Proceedings of the IEEE International Symposium on Intelligent Control, Intelligent Systems and Semiotics*. Boston, pp. 365–369.

Jones,S. and Thornton,J. (1995) Protein–protein interactions: a review of protein dimer structures. *Prog. Biophys. Mol. Biol.*, **63**, 31–65.

Kidera,A., Konishi,Y., Ooi,T. and Scheraga,H.A. (1985) Relation between sequence similarity and structural similarity in proteins. Role of important properties of amino acids. *J. Protein Chem.*, **4**, 265–297.

Klotz,I.M., Darnall,D.M. and Langerman,N.R. (1975) Quaternary structure of proteins. In Neurath,H. and Hill,R.L. (eds), *The Proteins*, vol 1. Academic Press, New York, pp. 293–411.

Larsen,T.A., Olson,A.J. and Goodsell,D.S. (1998) Morphology of protein–protein interfaces. *Structure*, **6**, 421–427.

Matthews,B.W. (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Nakashima,H., Nishikawa,K. and Ooi,T. (1990) Distinct character in hydrophobicity of amino acid compositions of mitochondrial proteins. *Proteins: Struct. Funct. Genet.*, **8**, 173–178.

Provost,F. and Fawcett,T. (1997) Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining KDD-97)*, pp. 43–48.

Provost,F. (2000) Learning with Imbalanced Data Sets 101. Invited paper for the *AAAI'2000 Workshop on Imbalanced Data Sets*.

Quinlan,J.R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

Quinlan,J.R. (2000) Personal communication.

Rost,B. and Sander,C. (1995) Progress of 1D protein structure prediction at last. *Proteins: Struct. Funct. Genet.*, **23**, 295–300.

Tomii,K. and Kanehisa,M. (1996) Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Eng.*, **29**, 27–36.

Tsai,C.-J., Lin,S.-L., Wolfson,H.J. and Nussinov,R. (1997) Studies of protein–protein interfaces: a statistical analysis of the hydrophobic effect. *Protein Sci.*, **6**, 53–64.

Weiss,S.M. and Kulikowski,C.A. (1990) *Computer Systems that Learn*. Morgan Kaufmann, San Mateo, CA.