Fall 2010

Protein secondary structure prediction using BLAST and relaxed threshold rule induction from coverings

Leong Lee

Department: Computer Science

PROTEIN SECONDARY STRUCTURE PREDICTION USING BLAST AND RELAXED THRESHOLD RULE INDUCTION FROM COVERINGS

by

LEONG LEE

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2010

Approved by

Dr. Jennifer L. Leopold, Advisor
Dr. Ronald L. Frank
Dr. Fikret Ercal
Dr. Ralph W. Wilkerson
Dr. Dan Lin

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following five articles that were published, or submitted for publication, as listed below. Each paper has been prepared in the style required by their respective journals/conferences.

Pages 12 to 31, “Protein Secondary Structure Prediction Using Rule Induction from Coverings” was published in the *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology 2009* (Paper 1, RT-RICO).

Pages 32 to 51, “Protein Secondary Structure Prediction Using Parallelized Rule Induction from Coverings” was published in *International Journal of Medicine and Medical Sciences,* Volume 1, Number 2, Spring 2010 (Paper 2, Parallelized RT-RICO).

Pages 52 to 83, “Protein secondary structure prediction using RT-RICO: a rule-based approach” was accepted for publication by *The Open Bioinformatics Journal* (Paper 3, Rule-based RT-RICO).

Pages 84 to 116, “Protein secondary structure prediction using BLAST and Relaxed Threshold Rule Induction from Coverings” was submitted to *BMC Bioinformatics* (Paper 4, BLAST-RT-RICO).

Pages 117 to 142, “Rule Visualization of Protein Motif Sequence Data for Secondary Structure Prediction” was accepted for publication by *ANNIE 2010 conference* (Paper 5, Rule Visualization).

ABSTRACT

Protein structure prediction has always been an important research area in bioinformatics and biochemistry. Despite the recent breakthrough of combining multiple sequence alignment information and artificial intelligence algorithms to predict protein secondary structure, the Q_3 accuracy of various computational prediction methods rarely has exceeded 75%; this status has changed little since 2003 when Rost stated that “the currently best methods reach a level around 77% three-state per-residue accuracy.”

The application of artificial neural network methods to this problem is revolutionary in the sense that those techniques employ the homologues of proteins for training and prediction. In this dissertation, a different approach, RT-RICO (Relaxed Threshold Rule Induction from Coverings), is presented that instead uses association rule mining. This approach still makes use of the fundamental principle that structure is more conserved than sequence. However, rules between each known secondary structure element and its “neighboring” amino acid residues are established to perform the predictions. This dissertation consists of five research articles that discuss different prediction techniques and detailed rule-generation algorithms. The most recent prediction approach, BLAST-RT-RICO, achieved a Q_3 accuracy score of 89.93% on the standard test dataset RS126 and a Q_3 score of 87.71% on the standard test dataset CB396, an improvement over comparable computational methods.

Herein one research article also discusses the results of examining those RT-RICO rules using an existing association rule visualization tool, modified to account for the non-Boolean characterization of protein secondary structure.

ACKNOWLEDGEMENTS

I would like to express my sincere and deepest gratitude to my advisor, Professor Jennifer L. Leopold. Her computer science knowledge, academic mentoring skills, and logical thinking have been of great value to me. Her understanding, encouraging character, and personal guidance ensured the timely delivery of this dissertation. I am fortunate to have had an advisor who allowed me to explore my own research ideas, and at the same time encouraged and guided me when I faced crisis.

I also am extremely grateful to Professor Ronald L. Frank, for his constructive contribution from the biological sciences point of view, and his important support and discussions. I am also indebted to Professor Anne M. Maglia for her professional advice and selfless assistance.

I wish to express my heartfelt thanks to the members of my advisory committee Professor Fikret Ercal, Professor Ralph W. Wilkerson, and Professor Dan Lin, for their valuable guidance, comments, feedback, and support. They definitely inspired me and helped to make this dissertation possible.

I am grateful to the following faculty members, graduate students, colleagues, and friends at Missouri S&T, in particular Professor Ali Hurson, Professor Bruce M. McMillin, Professor Sanjay Madria, Professor Sriram Chellappan, Professor Wei Jiang, Professor Xiaoqing (Frank) Liu, Professor Chaman L. Sabharwal, Professor Daniel R. Tauritz, Clayton Price, Rhonda Grayson, Dawn Davis, Waraporn Viyanon, Cyriac Kandoth, Alton B. Coalter, Patrick G. Edgett, Dr. Analia Pugener, Barbara Ann Fears, Bonnie Beasley, and Sarah Havens.

Not forgetting my church friends from First Presbyterian Church of Rolla, MO, I thank them for their hospitality and support throughout my stay in Rolla.

Most importantly, none of this would have been possible without the love and patience of my family. My capable wife Amy and my talented daughter Tabitha have been a constant source of love, support, understanding, and strength. My mother has aided me financially and emotionally. My mother-in-law has taken care of my daughter for me. My father and grandfather have encouraged and believed in me unconditionally.

Finally, I thank God for walking with me through this great learning journey.

TABLE OF CONTENTS

Page

SECTION

LIST OF ILLUSTRATIONS

PAPER1 Page

PAPER 2

PAPER 3

PAPER 4

PAPER 5

LIST OF TABLES

PAPER 1

PAPER 2

PAPER 3

PAPER 4

PAPER 5

1. INTRODUCTION

For many decades, the focus of computer science was on the fundamental theory of computation, which examined and studied various theoretical computation models, and the time and space costs associated with different computational solutions. In recent years, especially after the advent of the so-called "digital revolution", there has been an explosion of computer applications that involve other disciplines. This multidisciplinary approach to solving practical problems is a natural progression; interdisciplinary research fields can reap considerable benefit from both the theoretical, computer science viewpoint, as well as the more applied, domain-specific perspective. Bioinformatics is definitely one of the growing areas where computer science increasingly is being applied to another discipline. Solutions to numerous problems in genomics and proteomics require a fusion of methods from computer science, engineering, chemistry, and biology. Bioinformatics has gradually evolved to also entail the creation and design of databases, algorithms, statistical techniques, and theories to solve problems arising from the need for the management and analysis of vast amounts of heterogeneous biological data. The benefits of bioinformatics research are obvious; for example, research in proteomics and genetics has facilitated the creation of new medicines and the design of new enzymes. In the future, further developments can be expected to help us understand the intricacies of biological systems, and hence improve the quality of human life and the environment.

This Ph.D. dissertation focuses on an important proteomics research problem, protein secondary structure prediction. Prediction of the 3D structure of a protein from its amino acid sequence is a very important and challenging research goal in bioinformatics, and has been studied extensively since the 1960s. Rost (2003) suggested that protein 3D structure prediction from sequence still cannot be achieved fully. However, research has continuously improved computational methods for predicting simplified aspects of structure.

Despite the recent breakthrough of combining multiple sequence alignment information and artificial intelligence algorithms to predict protein secondary structure, the Q_3 accuracy of various computational prediction methods rarely has exceeded 75%;

this status has changed little since 2003 when Rost stated that “the currently best methods reach a level around 77% three-state per-residue accuracy.”

This dissertation contains five research papers that were published, or have been submitted for publication and are currently in review. The research was done under the supervision of Dr. Jennifer L. Leopold and Dr. Ronald L. Frank, from the Missouri University of Science & Technology (Missouri S&T) Computer Science and Biological Sciences Departments, respectively. In the first paper (Paper 1, RT-RICO), a newly developed rule-based data-mining approach called RT-RICO (Relaxed Threshold Rule Induction from Coverings) is presented. This method identifies dependencies between amino acids in a protein sequence, and generates rules that can be used to predict secondary structure. RT-RICO uses some of the concepts introduced by Pawlak (1984) for rough sets, a classification scheme based on partitions of entities in a dataset (Grzymala-Busse, 1991). Four new definitions and two new algorithms are presented to form the main RT-RICO rule generation algorithm. The average prediction accuracy, or Q_3 score, on a non-standard test dataset was 80.3% (Lee, Leopold, Frank and Maglia, 2009).

For the second paper (Paper 2, Parallelized RT-RICO), a parallelized implementation of a slightly modified RT-RICO approach is presented; Cyriac Kandoth, a recently graduated Ph.D. student from the Missouri S&T Computer Science Department is responsible for the design and implementation of the parallelized rule-generation algorithm. This new version of algorithm, with an improved time complexity, facilitated the testing of a much larger standard test dataset, CB396. Parallelized RT-RICO achieved a Q_3 score of 74.6% (Lee, Kandoth, Leopold and Frank, 2010a).

The third paper (Paper 3, Rule-based RT-RICO) discusses further improvements to the prediction algorithm, which resulted in a more accurate prediction on standard test datasets. RT-RICO achieved a Q_3 score of 81.75% on the standard test dataset RS126, and a Q_3 score of 79.19% on the standard test dataset CB396, both of which were improvements over comparable computational methods (Lee, Leopold, Kandoth and Frank, 2010b).

For the fourth paper of this dissertation (Paper 4, BLAST-RT-RICO), a modified method for predicting the secondary structure elements, BLAST-RT-RICO, is presented.

First, a query using the Web-based NCBI/PSI-BLAST search engine is performed for a protein (BLAST, 2009). Suitable proteins with significant multiple sequence alignments are identified. Then the RT-RICO algorithm is used to generate rules representing dependencies between protein amino acid sequences and the related secondary structure elements. The BLAST-RT-RICO method performed better than our previously developed method, with a Q_3 accuracy of 89.93% on the RS126 set and 87.71% on the CB396 set (Lee, Leopold and Frank, 2010c).

The success of the rule-based methods supports the belief that there are meaningful statistical relationships between any secondary structure position and its neighboring amino acids. However, because of the vast amount of rules generated by RT-RICO, potentially useful information within the rule set can be difficult to identify. In the fifth paper (Paper 5, Rule Visualization), modifications to an existing visualization technique are proposed in order to analyze the association rules. This technique not only enables users to visualize the rules, but also allows users to compare rule sets between different protein classes, and to compare rule sets of different test proteins (Lee, Leopold, Edgett and Frank, 2010d).

2. REVIEW OF LITERATURE

2.1. PROTEIN SECONDARY STRUCTURE PREDICTION

Protein secondary structure prediction aims to predict the secondary structure of proteins based on knowledge of their primary structure, amino acid sequence.

Prediction of the 3D structure of a protein from its amino acid sequence is a very important research goal in biochemistry and bioinformatics. Rost (2003) suggests that although protein 3D structure prediction from sequence cannot be achieved fully, in general, research has continuously improved methods for predicting simplified aspects of structure. Particularly in the area of secondary structure prediction, accuracy has surpassed the 70% threshold for all residues of a protein. That breakthrough was achieved by combining multiple sequence alignment information and artificial intelligence algorithms. Rost (2003) also has stated that a value of around 88% likely will be the operational upper limit for prediction accuracy.

Kabsch and Sander developed a set of simple and physically motivated criteria for secondary structure, programmed as a pattern-recognition process of hydrogen-bonded and geometrical features extracted from x-ray coordinates (Kabsch and Sander, 1983). This DSSP (Define Secondary Structure of Proteins) algorithm is the standard method for assigning secondary structure to the primary structure (amino acids) of a protein. Depending on the pattern of hydrogen bonds, DSSP recognizes eight types or states of secondary structure. The 3-helix (3/10 helix), alpha helix, and 5 helix (pi helix) are symbolized as G, H and I, respectively. DSSP recognizes two types of hydrogen-bond pairs in beta sheet structures, the parallel and antiparallel bridge. A residue in isolated beta-bridge is symbolized by B, whereas E represents an extended strand, and participates in a beta ladder. The remaining types are T for hydrogen bonded turn, and S for bend. There is also blank or “-” meaning “loop” or “other.” These eight types are usually grouped into three classes: helix (G, H, and I), strand/sheet (E and B) and loop/coil (all others).

Given the atomic-resolution coordinates of a protein, the standard method for assigning secondary structure to the amino acids is the DSSP algorithm. However, the experimental methods used to determine the structures of proteins demand sophisticated equipment and time (Fadime, O¨zlem and Metin, 2008). As a result, many computational methods are developed to predict the location of secondary structure elements in proteins for complementing or creating insights into experimental results.

2.2. PROBLEM DESCRIPTION

In general, the protein secondary structure prediction problem can be characterized in terms of the following components (Baldi et al., 2000):

Input

Amino acid sequence, $A = a_1, a_2, \ldots a_N$

Data for comparison, $D = d_1, d_2, \ldots d_N$

a_i is an element of a set of 20 amino acids, $\{A,R,N\ldots V\}$

d_i is an element of a set of secondary structures, $\{H,E,C\}$, which represents helix H, sheet E, and coil C.

Output

Prediction result: $X = x_1, x_2, \ldots x_N$

x_i is an element of a set of secondary structures, $\{H,E,C\}$

3-Class Prediction (Zhang and Zhang, 2003)

This is a characterization of the problem as a multi-class prediction problem with 3 classes $\{H,E,C\}$ in which one obtains a 3 × 3 confusion matrix $Z = (z_{ij})$. z_{ij} represents the number of times the input is predicted to be in class j while belonging to class i.

$Q_{\text{total}} = 100 \sum_i Z_{ii} / N$

Q_3 Score

Accuracy is computed as $Q_3 = W_{\alpha\alpha} + W_{\beta\beta} + W_{cc}$

$W_{\alpha\alpha}$ = % of helices correctly predicted ($100\ Z_{11} / N$ or $100\ Z_{HH} / N$)

$W_{\beta\beta}$ = % of sheets correctly predicted ($100\ Z_{22} / N$ or $100\ Z_{EE} / N$)

W_{cc} = % of coils correctly predicted ($100\ Z_{33} / N$ or $100\ Z_{CC} / N$)

In other words, a protein secondary structure data sequence D is compared to the predicted result sequence X to calculate the Q_3 accuracy score.

2.3. THREE GENERATIONS OF PREDICTION METHODS

Rost (2003) classifies protein secondary structure prediction methods into three generations. The first generation methods depend on single residue statistics to perform prediction. The second generation methods depend on segment statistics. The third generation methods use evolutionary information to predict secondary structure. For example, PHD (Rost and Sander, 1993a) is a third generation prediction method based on a multiple-level neural network approach. It has been the most accurate method for many years.

Many third generation prediction methods use similar neural network approaches. These artificial neural network methods are revolutionary in the sense that they employ the homologues of proteins for training and prediction. In PHD (Rost and Sander, 1993a), Rost and Sander use multiple sequence alignments rather than single sequences as input to a neural network. At the training stage, a database of protein families aligned to proteins of known structure is used. At the prediction stage, the database of sequences is scanned for all homologues of the protein to be predicted, and the family profile of amino acid frequencies at each alignment position is fed into the network (Rost and Sander, 1993b).

A key consideration in many of the third generation methods is the knowledge that random mutations in DNA sequence can lead to different amino acids in the protein sequences. These changes are considered the basis of evolution; mutations resulting in a structural change are not likely to retain protein function. Thus, structure is more conserved than sequence (Rost, 2003). All naturally evolved protein pairs that have 35 of 100 pairwise identical residues have similar structures (Rost, 2003). This is the basis of how evolutionary information is used in the form of multiple sequence alignments for predicting protein secondary structure.

It is not an easy task to evaluate the performance of a protein secondary structure prediction method. For example, the use of different datasets for training and testing each algorithm makes it difficult to find an objective comparison of methods (Cuff and Barton, 1999). Rost (2003) stated that “there is no value in comparing methods evaluated on different datasets.” Efforts have been made to develop standard test datasets to accurately evaluate the performance of prediction methods. Rost and Sander (1993a) selected a list of 126 protein domains (the RS126 set) that now constitutes a comparative standard. Cuff and Barton (1999) described the development of a non-redundant test set of 396 protein domains (the CB396 set) where no two proteins in the set share more than 25% sequence identity over a length of more than 80 residues (Rost and Sander, 1993a). They used the CB396 set to test four secondary structure prediction methods: PHD (Rost and Sander, 1993a), DSC (King and Sternberg, 1996), PREDATOR (Frishman and Argos, 1997) and NNSSP (Salamov and Solovyev, 1995). They also combined the four methods by a simple majority-wins method, the CONSENSUS method (Cuff and Barton, 1999). The resulting Q_3 scores for the CB396 set were 71.9% (PHD), 68.4% (DSC), 68.6% (PREDATOR), 71.4% (NNSSP) and 72.9% for the CONSENSUS method. In the same research study, Cuff and Barton (1999) also tested the RS126 set in which the Q_3 scores were 73.5% (PHD), 71.1% (DSC), 70.3% (PREDATOR), 72.7% (NNSSP) and 74.8% for the CONSENSUS method; see Table 2.1 for an overview of Q_3 scores of secondary structure prediction methods.

Recently, there has been a trend to use the support vector machine (SVM) to predict protein secondary structures. Hu, Pan, Harrison and Tai (2004) achieved a Q_3 accuracy of 78.8% on the RS126 dataset using a SVM approach. Kim and Park (2003) developed the SVMpsi method that resulted in Q_3 scores of 76.1% on the RS126 dataset and 78.5% on their KP480 dataset. Nguyen and Rajapakse (2007) proposed a two-stage multi-class SVM approach utilizing position-specific scoring matrices generated by PSI-BLAST; the resulting Q_3 scores were 78.0% on the RS126 dataset and 76.3% on the CB396 dataset.

Table 2.1 - Q_3 Scores of Secondary Structure Prediction Methods

Methods	RS126 Test Dataset	CB396 Test Dataset	Other Test Datasets
PHD (Rost and Sander, 1993a)	73.5%	71.9%	
DSC (King and Sternberg, 1996)	71.1%	68.4%	
PREDATOR (Frishman and Argos, 1997)	70.3%	68.6%	
NNSSP (Salamov and Solovyev, 1995)	72.7%	71.4%	
CONSENSUS (Cuff and Barton, 1999)	74.8%	72.9%	
Fadime, 2-stage (Fadime, O¨zlem and Metin, 2008)			74.1%
PSIPRED (Jones, 1999)			78.3%
Hu, SVM (Hu, Pan, Harrison and Tai, 2004)	78.8%		
Kim, SVMpsi (Kim and Park, 2003)	76.1%		78.5%
Nguyen, 2-stage SVM (Nguyen and Rajapakse, 2007)	78.0%	76.3%	
BLAST-RT-RICO	89.9%	87.7%	

Note: Due to the different approaches, different protein secondary structure data availability and different test design strategies, it is difficult to directly compare different methods' prediction results. The Q_3 scores comparison should be used as a general guide, not a strict percentile comparison.
Q_3 scores of PHD (Rost and Sander, 1993a), DSC (King and Sternberg, 1996), PREDATOR (Frishman and Argos, 1997) and NNSSP (Salamov and Solovyev, 1995) are from the research paper of Cuff and Barton (1999).
Q_3 scores under "Other Test Datasets" column should NOT be directly compared, because they use different test datasets.

PAPER

1. PROTEIN SECONDARY STRUCTURE PREDICTION USING RULE INDUCTION FROM COVERINGS

Leong Lee, Jennifer L. Leopold, Ronald L. Frank and Anne M. Maglia

Leong Lee and Jennifer L. Leopold are affiliated with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: {llkr4, leopoldj}@mst.edu).

Ronald L. Frank and Anne M. Maglia are affiliated with the Department of Biological Sciences, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: {rfrank, magliaa}@mst.edu).

Abstract— With the increase of data from genome sequencing projects comes the need for reliable and efficient methods for the analysis and classification of protein motifs and domains. Experimental methods currently used to determine protein structure are accurate, yet expensive both in terms of time and equipment. Therefore, various computational approaches to solving the problem have been attempted, although their accuracy has rarely exceeded 75%. In this paper, a rule-based method to predict protein secondary structure is presented. This method uses a newly developed data-mining algorithm called RT-RICO (Relaxed Threshold Rule Induction from Coverings), which identifies dependencies between amino acids in a protein sequence, and generates rules that can be used to predict secondary structures. The average prediction accuracy on sample data sets, or Q_3 score, using RT-RICO was 80.3%, an improvement over comparable computational methods.

I. INTRODUCTION

Developing or identifying methods to discover patterns in protein sequences, and thus identifying protein structure, is one of the most challenging problems in computational genomics. Experimental determination of protein structures using Nuclear Magnetic Resonance spectroscopy (NMR) or X-ray crystallography are accurate, yet time consuming and expensive. Thus, protein structure predictions often are made using computational methods. However, current ab initio methods that predict protein structures from amino acid sequences are computationally demanding, and currently are limited to relatively small proteins with short amino acid sequences [1]. Furthermore, large amounts of computer time and resources are required to build structure models for each newly discovered protein sequence.

Many studies have attempted to develop computational methods to predict protein motif structure from empirical data. One of the best such structure predictors is Jones' PSIPRED Protein Structure Prediction Server, which was developed at University College London [2], [3]. PSIPRED uses a two-stage neural network to predict the protein's secondary structure based on position-specific scoring matrices. The matrices are generated by PSI-BLAST (Position-Specific Iterated BLAST) [4], which automatically combines statistically significant alignments produced by BLAST into a matrix, and then searches the database using the values in the matrix. PSIPRED makes its predictions with an average accuracy, or Q_3, score of between 76.5% and 78.3% [2]. A number of other secondary structure predictors also utilize a neural network prediction algorithm. One of these systems, Jnet, works by applying multiple sequence alignments alongside profiles such as PSI-BLAST and HMM [5].

Another interesting structure prediction method was presented by Fadime, O¨zlem, and Metin [7]. It used a two-stage method to predict the protein secondary structure. In the first stage the folding type of a protein is determined. The second stage utilizes data from the Protein Data Bank (PDB) and a probabilistic search algorithm to determine the locations of secondary structure elements. The resulting average accuracy of their prediction score is 74.1%.

In this paper, we present a more accurate method for predicting the secondary structure elements for each folding type. Our algorithm, RT-RICO (Relaxed Threshold

Rule Induction from Coverings), generates rules for discovering non-independent patterns between protein amino acid sequences and related secondary structure elements. These rules are then used to predict protein secondary structure.

The results of this method are presented in Section IV, and the RT-RICO algorithm is discussed in detail in Section V.

II. PROBLEM DESCRIPTION

In general, the protein secondary structure prediction problem can be characterized in terms of the following components [8]:

- Input

Amino acid sequence, $A = a_1, a_2, \ldots a_N$

Data for comparison, $D = d_1, d_2, \ldots d_N$

a_i is an element of a set of 20 amino acids, $\{A,R,N\ldots V\}$

d_i is an element of a set of secondary structures, $\{H,E,C\}$, which represents helix H, sheet E, and coil C.

- Output

Prediction result: $M = m_1, m_2, \ldots m_N$

m_i is an element of a set of secondary structures, $\{H,E,C\}$

- 3-Class Prediction [9]

This is a characterization of the problem as a multi-class prediction problem with 3 classes $\{H,E,C\}$ in which one obtains a 3 x 3 confusion matrix $Z = (z_{ij})$. z_{ij} represents the number of times the input is predicted to be in class j while belonging to class i.

$Q_{\text{total}} = 100 \sum_i Z_{ii} / N$

- Q_3 Score

Accuracy is computed as $Q_3 = W_{\alpha\alpha} + W_{\beta\beta} + W_{cc}$

$W_{\alpha\alpha}$ = % of helices correctly predicted

$W_{\beta\beta}$ = % of sheets correctly predicted

W_{cc} = % of coils correctly predicted

In other words, a protein secondary structure data sequence D is compared to the prediction result sequence M to calculate the Q_3 score.

III. RELATED WORK

Levitt and Chotia proposed to classify proteins as four basic types according to their α-helix and β-sheet content [10]. “All-α” class proteins consist almost entirely (at least 90%) of α-helices. “All-β” class proteins are composed mostly of β-sheets (at least 90%). The “α/β” class proteins have alternating, mainly parallel segments of α-helices and β-sheets. The “α+β” class proteins have a mixture of all-α and all-β regions, mostly in sequential order. Fadime, O¨zlem, and Metin developed a two-stage method to predict secondary structure of proteins [7]. In the first stage of their method, they are able to determine the class of unknown proteins with 100% accuracy. Given a protein sequence, they use a mixed-integer linear program (MILP) approach to decide if the protein sequence belongs to one of the four classes (“all-α”, “all-β”, “α/β”, or “α+β”). In the second stage of their method, they use a probability approach based on their stage one results. They decompose the amino acid sequences of the training set into overlapping sequence groups of three to seven residues. These groups are used to calculate the probability statistics for secondary structure. Specifically, the secondary structure at a particular sequence location is determined by comparing the probabilities that an amino acid residue is a particular secondary structure type based on the statistics.

Their results are impressive. They achieved a 100% accuracy for classifying proteins into one of the four protein type classes (“all-α”, “all-β”, “α/β”, or “α+β”). This greatly simplifies part of the protein secondary structure prediction problem. That is, given a protein amino acid sequence, if we know which one of the four classes this protein belongs to, we can apply other approaches to predict the secondary structure elements within these four classes. In contrast, our method, RT-RICO, (discussed in more detail in section V) uses a rule-based approach as an alternative way to make the prediction.

Other studies have also tried to identify patterns within an amino acid sequence. Wang, Schroeder, Dobbs, and Honavar investigated a data-driven approach to the discovery of rules for assigning protein sequences to functional families on the basis of the presence or absence of specific motifs or combinations of motifs [18]. They mapped each protein sequence into a corresponding attribute-based representation, and used a learning algorithm to assign novel protein sequences to one of the protein families represented in the training set. In later work, Wang et al., developed an algorithm to find patterns in 3D graphs in order to locate frequently occurring motifs in two families of proteins, and then used the motifs to classify the proteins [19]. Davey, Shields, and Edwards also addressed the identification problem by establishing methods for discovering putative functional motifs occurring in unrelated proteins that evolve by convergence [20].

A study by Maglia, Leopold and Ghatti [21] implemented a data mining approach based on rule induction from coverings in order to identify non-independence in phylogenetic data. For such data sets, this approach was shown to be preferable over two other commonly used approaches for representing data dependencies in terms of rules: (1) Bayesian analysis (which is dependent upon an ordering of attributes in the data set), and (2) decision tree induction (which only produces a partial set of rules, none of which is necessarily correct for all instances in the data set). Although rule induction from coverings appeared to be a promising solution for the phylogenetic data non-independence problem, it suffered from exponential computational complexity (which was in part addressed by a parallelized implementation by Leopold et al. [22]), as well as the strictness required for the resulting rules (i.e., all rules had to be correct for all instances in the data set). In addition, the restrictive requirements for the rules impeded the discovery of meaningful relationships in the phylogenetic data sets, as well as in protein data sets. Rather than abandoning the rule induction from coverings approach altogether, we decided to try relaxing the restrictive requirements for the rules, as is discussed in the next section.

IV. RESULTS

We believe that it will be easier for the reader to understand the method if s/he first fully understands what we are trying to achieve. Therefore, before explaining the details of how RT-RICO works, we will present the results of our tests.

As test data, protein names and corresponding folding types of each protein were obtained from the SCOP database [11], [12]. The protein sequences and secondary structure sequences were retrieved from the PDB database [13]. We built four databases of proteins (with their amino acid sequences and secondary structure sequences) of different protein types ("all-α", "all-β", "α/β", and "α+β"). We selected proteins from different protein families to form the training data sets and the test data sets. See Table I for the number of proteins in each training data set.

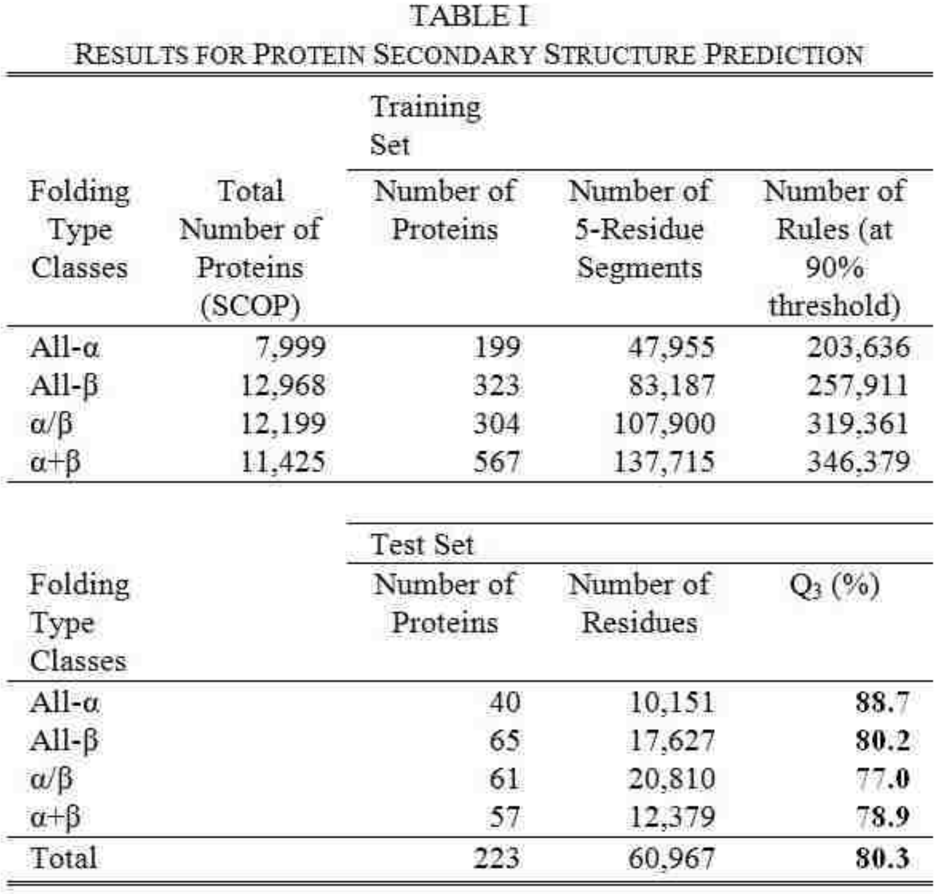


TABLE I

RESULTS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

		Training Set		
Folding Type Classes	Total Number of Proteins (SCOP)	Number of Proteins	Number of 5-Residue Segments	Number of Rules (at 90% threshold)
All-α	7,999	199	47,955	203,636
All-β	12,968	323	83,187	257,911
α/β	12,199	304	107,900	319,361
α+β	11,425	567	137,715	346,379

	Test Set		
Folding Type Classes	Number of Proteins	Number of Residues	Q_3 (%)
All-α	40	10,151	**88.7**
All-β	65	17,627	**80.2**
α/β	61	20,810	**77.0**
α+β	57	12,379	**78.9**
Total	223	60,967	**80.3**

For the first three classes ("all-α", "all-β", and "α/β"), approximately 2.5% of all the available proteins (from SCOP) were chosen as training data. For the "α+β" class, approximately 5% of all the available proteins were chosen as training data. We chose 5% for the last class mainly because we wanted to have enough 5-residue segments for the "α+β" class. If we used only 2.5%, the number of 5-residue segments for the "α+β" class would be much less than that for the "α/β" class. The PDB Ids for all protein sequences used for training and testing can be found on the following webpage: http://www.leeleong.com/rt-rico/.

The protein secondary structure sequences from PDB are formed by elements of eight states of secondary structure, {H, G, I, E, B, T, S, -}. The eight states were converted to four states to facilitate rule generation as follows:

(G, H, I) => Helix H

(E, B) => Sheet E

(T, S) => Coil C

(-) => "-"

Note that rule generation uses a four-state decision attribute. The final Q_3 score calculation uses a three-state decision attribute:

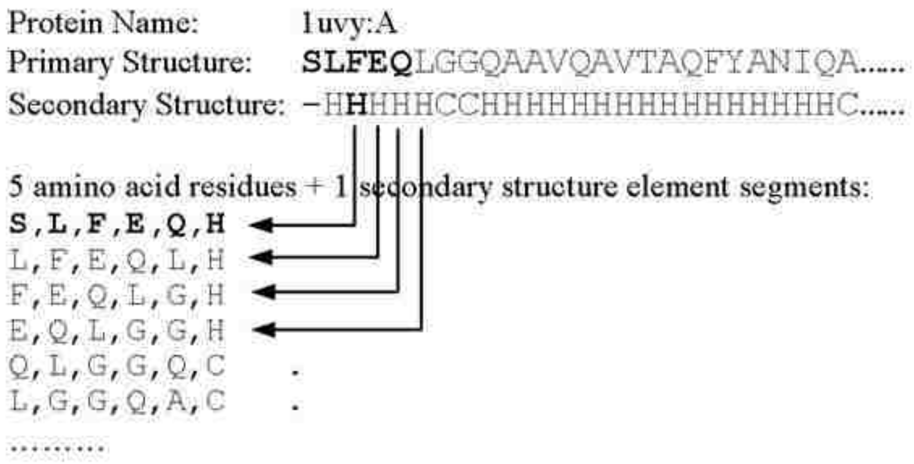
(G, H, I) => Helix H

(E, B) => Sheet E

(Rest) => Coil C

The basis for our approach is to first search segments of amino acid sequences of known protein secondary structures, and then find the rules that relate amino acid residues to secondary structure elements. The generated rules subsequently are used to predict the secondary structure. Klepeis and Floudas showed that the use of overlapping segments of five residues is very effective in predicting the helical segments of proteins [14]. Thus, we used the overlapping 5-residue segments approach to prepare the training data records. As shown in Fig. 1, for each secondary structure element, five "neighboring" amino acid residues were extracted to form a segment of five amino acid residues, plus one secondary structure element. These segments were used as input to the RT-RICO algorithm to generate rules. The numbers of 5-residue segments generated for the four protein type classes are shown in Table I.

The inputs to the RT-RICO are in the form of a 6-tuple. The first five elements of the 6-tuple are formed by amino acid residues, {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. The last element of the 6-tuple is formed by one of four secondary structure states {H, E, C, -}. The last element is the decision attribute.



Protein Name: 1uvy:A

Primary Structure: SLFEQLGGQAAVQAVTAQFYANIQA......

Secondary Structure: –HHHHCCHHHHHHHHHHHHHHHHHC......

5 amino acid residues + 1 secondary structure element segments:

S,L,F,E,Q,H

L,F,E,Q,L,H

F,E,Q,L,G,H

E,Q,L,G,G,H

Q,L,G,G,Q,C .

L,G,G,Q,A,C .

....

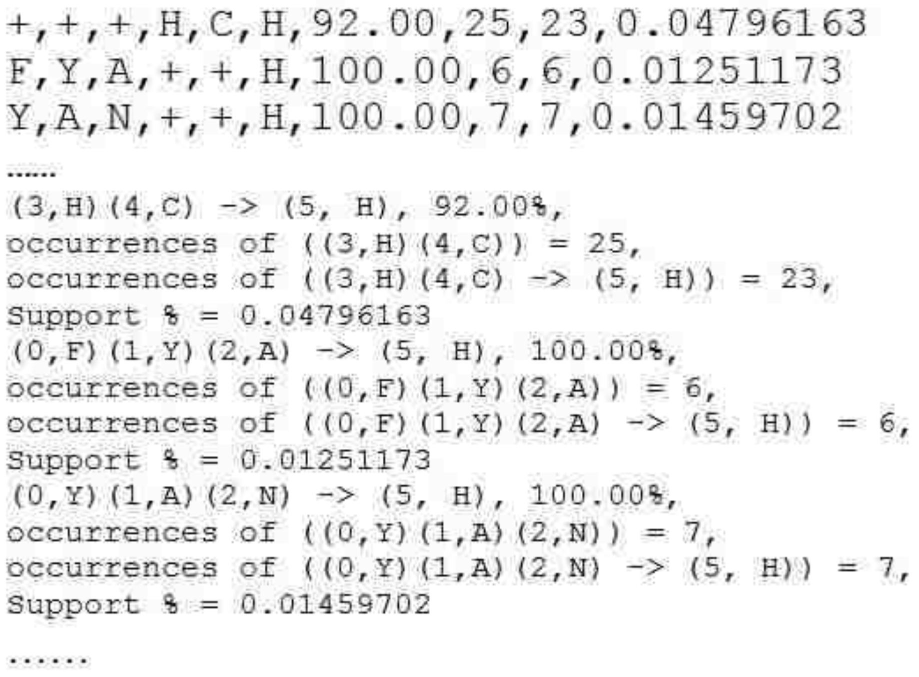
Note: The first and second positions at the beginning of the sequence are represented by 3 residues + 1, and 4 residues + 1 segments respectively. They form separate training data sets.

Fig. 1. Protein primary structure 5-residue segments and related secondary structure elements representation.

RT-RICO generated rules based on the segments. Some examples of these rules are shown in Fig. 2, in two separate formats. The first format is to be read by the computer programs at the later prediction stage (computer rule format). The second format is to be read by the user (human rule format). The first rule (in human rule format) is interpreted as if the fourth position attribute (or “3” interpreted by program) is “H”, and the fifth position attribute (or “4” interpreted by program) is “C”, then the sixth attribute (decision attribute, or “5” interpreted by program) is “H”, the confidence is 92%, and the support is 0.04796163%. The definitions of confidence and support can be found in [23].

The corresponding first rule (in computer rule format) is interpreted as if the first position attribute is “+” (represents any amino acid element), the second position attribute is “+”, the third position attribute is “+”, the fourth position attribute is “H”, and the fifth position attribute is “C”, then the sixth attribute (decision attribute) is “H”. The number of occurrences of the fourth position attribute is “H”, the fifth position attribute is “C”, and the sixth attribute is “H”, equals 25 among all inputs to RT-RICO. The number of occurrences of the fourth position attribute is “H”, and the fifth position attribute is “C”, equals 23 among all inputs to RT-RICO. The support is 0.04796163%.

Finally RT-RICO loads protein primary structures from the test data set, and predicts the secondary structure elements. As shown in Fig. 3, for each secondary structure element prediction position, five "neighboring" amino acid residues were extracted to form a segment of five amino acid residues. Each of these segments was compared with the generated rules. If a segment matched a rule, the support value of the rule was taken into consideration for the prediction of the related secondary structure element. We first searched for matching rules with 100% confidence value. If no matching rule existed among 100% confidence value rules, we then searched for other rules for matches. The secondary structure element with the highest total support value was selected as the predicted secondary structure element for the specific position. The number of proteins used in the test data sets, and the final Q_3 scores are shown in Table I.

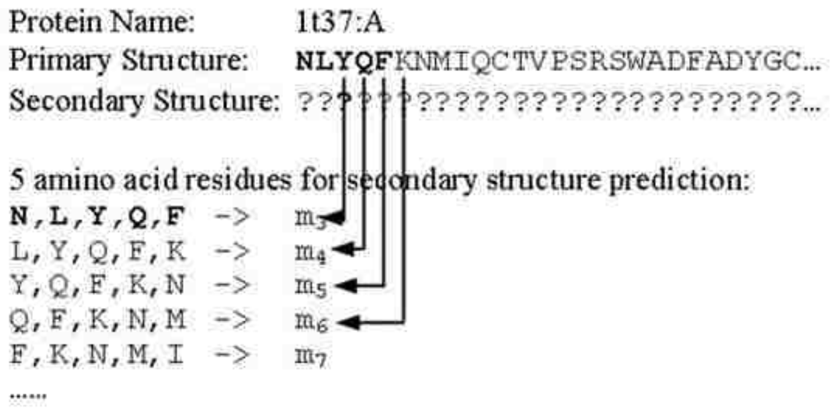


```
+,+,+,H,C,H,92.00,25,23,0.04796163
F,Y,A,+,+,H,100.00,6,6,0.01251173
Y,A,N,+,+,H,100.00,7,7,0.01459702
......
(3,H)(4,C) -> (5, H), 92.00%,
occurrences of ((3,H)(4,C)) = 25,
occurrences of ((3,H)(4,C) -> (5, H)) = 23,
Support % = 0.04796163
(0,F)(1,Y)(2,A) -> (5, H), 100.00%,
occurrences of ((0,F)(1,Y)(2,A)) = 6,
occurrences of ((0,F)(1,Y)(2,A) -> (5, H)) = 6,
Support % = 0.01251173
(0,Y)(1,A)(2,N) -> (5, H), 100.00%,
occurrences of ((0,Y)(1,A)(2,N)) = 7,
occurrences of ((0,Y)(1,A)(2,N) -> (5, H)) = 7,
Support % = 0.01459702
......
```

Fig. 2. Sample rules generated by RT-RICO.

The "all-α" proteins have the highest Q_3 score of 88.7%. The "all-β" and "α+β" proteins have Q_3 scores of 80.2% and 78.9%, respectively. The "α/β" proteins have the lowest prediction accuracy of 77.0%.

The test programs (rule-generation and prediction for four classes) were written in PERL and executed on a computer with Intel Pentium Dual-Core processor, 2 GB of RAM, and Windows XP OS. The total program running time was approximately 14 days.



Protein Name: 1t37:A
Primary Structure: **NLYQF**KNMIQCTVPSRSWADFADYGC...
Secondary Structure: ?????????????????????????...

5 amino acid residues for secondary structure prediction:
N,L,Y,Q,F -> m_3
L,Y,Q,F,K -> m_4
Y,Q,F,K,N -> m_5
Q,F,K,N,M -> m_6
F,K,N,M,I -> m_7
......

Fig. 3. Protein primary structure 5-residue segments and related secondary structure elements prediction. m_i is an element of set {H,E,C,-}. It is then converted to an element of set {H, E, C}. Note: The first and second positions at the beginning of the sequence are represented (predicted) by 3 residue, and 4 residue segments respectively. Their related prediction is handled slightly differently.

V. RT-RICO ALGORITHM

A. Rule Induction From Coverings

RT-RICO is based on a previously implemented method called RICO (Rule Induction From Coverings) [21]. RICO uses some of the concepts introduced by Pawlak for rough sets, a classification scheme based on partitions of entities in a data set [15], [16].

In this approach, if *S* is a set of attributes and *R* is a set of decision attributes (i.e., attributes whose values we are interested in being able to determine if the values of the attributes in the set *S* are known), then a covering *P* of *R* in *S* can be found if the following three conditions are satisfied:

i. *P* is a subset of *S*.

ii. *R depends* on *P* (i.e., *P determines R*). That is, if a pair of entities *x* and *y* cannot be distinguished by means of attributes from *P*, then *x* and *y* also cannot be distinguished by means of attributes from *R*. If this is true, then entities *x* and *y* are said to be *indiscernible* by *P* (and, hence, *R*), denoted $x \sim_P y$. An *indiscernibility relation* $\sim_P$ is such a partition over all entities in the data set.

iii. *P* is minimal.

Condition (ii) is true if and only if an equivalent condition ≤, known as the *attribute dependency inequality*, holds for *P** and *R**, the partitions of all attributes and decisions generated by *P* and *R*, respectively, where, for a set of attributes *A*:

$$A^* = \prod_{a \in A} \sim [a]^*$$

The inequality $P^* \leq R^*$ holds if and only if for each block *B* of *P**, there exists a block *B′* of *R** such that *B* is a subset of *B′*.

Once a covering is found, it is a straightforward process to induce rules from it. For example, if a set of attributes $P = \{a_1, a_2\}$ is found to determine a set of attributes $R = \{a_3\}$ (i.e., *P* is a covering for *R*), then rules of the form $(a_1, v_1) \wedge (a_2, v_2) \rightarrow (a_3, v_3)$ (read as “if a_1 equals v_1 and a_2 equals v_2, then a_3 equals v_3) can be generated where v_1, v_2, and v_3 are actual values of attributes a_1, a_2, and a_3, respectively, for which the relationship holds in the data set. Such a rule also conveys a notion of non-independence between the attributes in the sets *P* and *R* (e.g., a_3 is non-independent of a_1 and a_2).

B. Relaxed Attribute Dependency Inequality

All rules generated from coverings in this manner are “perfect” in the sense that there is no instance in the data set for which the rule is not true. In order to relax this restriction somewhat (in much the same way that rules generated by decision tree induction are not always true for all instances in the data set), the definition of the attribute dependency inequality can be modified as follows.

Definition 1: Relaxed Attribute Dependency Inequality

The inequality $P^* \leq_r R^*$ holds if and only if *there exists* a block *B* of *P**, and *there exists* a block *B′* of *R** such that *B* is a subset of *B′*.

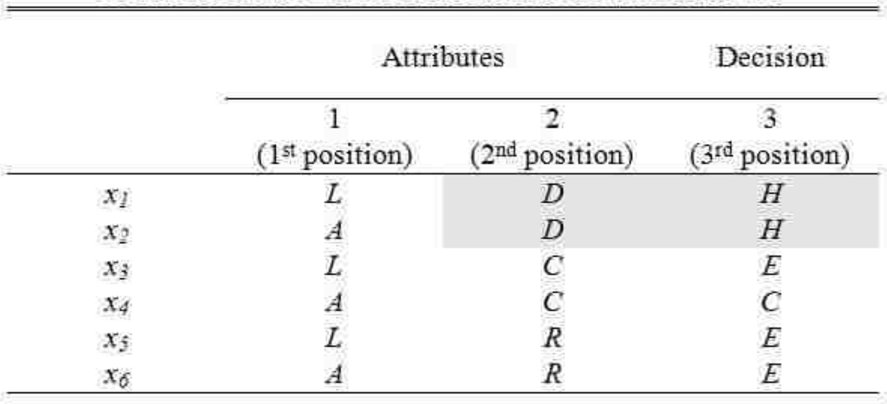
As an example for the data set of Table II, let $P = \{2\}$ and $R = \{3\}$. Then

$\{2\}^* = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6\}\}$

$\{3\}^* = \{\{x_1, x_2\}, \{x_3, x_5, x_6\}, \{x_4\}\}$

There exists a block $B = \{x_1, x_2\}$ in $\{2\}^*$ and a block $B' = \{x_1, x_2\}$ in $\{3\}^*$ such that $B \subseteq B'$. Thus, $\{2\}^* \leq_r \{3\}^*$ which means that $\{3\}$ depends on $\{2\}$ (i.e., $\{2\} \rightarrow_r \{3\}$) for at least *some* values of $\{2\}$. More specific rules can then be deduced from this relationship, such as $(2, D) \rightarrow (3, H)$.

TABLE II
DECISION TABLE WITH INDISCERNIBLE RELATIONSHIPS



	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*L*	*D*	*H*
x_2	*A*	*D*	*H*
x_3	*L*	*C*	*E*
x_4	*A*	*C*	*C*
x_5	*L*	*R*	*E*
x_6	*A*	*R*	*E*

C. Relaxed Coverings

Similarly, we can relax the definition of a covering in order to be able to induce rules depending on as small a number of attributes as possible.

Definition 2: Relaxed Coverings

A subset P of the set S is called a *relaxed covering* of R in S if and only if $P \rightarrow_r R$ and P is minimal in S. This is equivalent to saying that a subset P of the set S is a *relaxed covering* of R in S if and only if $P \rightarrow_r R$ and no proper subset P' of P exists such that $P' \rightarrow_r R$.

As an example for the data set of Table II, suppose we want to induce rules for $R = \{3\}$. The covering $\{1, 2\}$ can be used; that is, for any assignment of values for the covering $\{1, 2\}$, each entity in Table II will induce a rule for $\{3\}$. But, instead of inducing a rule from looking at combinations of values for $\{1, 2\}$, such as $(1, L) \wedge (2, D) \rightarrow (3, H)$, we will induce rules based on values for only *{1}* or *{2}*. Thus, $(2, D) \rightarrow (3, H)$ will be generated as a rule since $\{2\} \rightarrow_r \{3\}$ and *{2}* is minimal in *{1, 2}*. In this manner, *{2}* is a *relaxed covering* of *{3}*.

D. Checking Attribute Dependency

To implement rule induction from coverings with the relaxed constraints, it is necessary to use the concept of checking attribute dependency, which was introduced by Grzymala-Busse [16]. In order for *P* to be a relaxed covering of *R* in *S*, the following conditions must be true:

i. *P* must be a subset of *S*,
ii. *R* must depend on set *P* (for some values of *P*), and
iii. *P* must be minimal.

For our specific application, to generate rules for protein secondary structure prediction, rules involving more attributes are preferred over rules involving fewer attributes, because they normally generate higher confidence values. In addition, we need all the possible attribute position combinations. As a result, condition (iii) is not enforced for rule generation in our implementation.

Condition (ii) is true if and only if the relaxed attribute dependency inequality, $P^* \leq_r R^*$, is satisfied.

The question is then how do we efficiently check the above inequality? For each set *P*, a new partition, generated by *P*, must be determined. Partition *U* should be generated by *P*. For partitions π and τ of *U*, $\pi \cdot \tau$ is a partition of *U* such that two entities, *x* and *y*, are in the same block of $\pi \cdot \tau$ if and only if *x* and *y* are in the same block for both partitions π and τ of *U*. For example, referring to Table III,

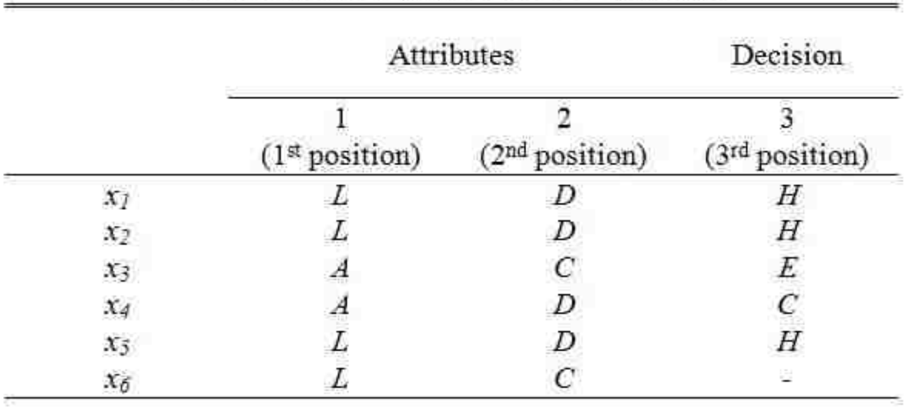
$\{1\}^* = \{\{x_1, x_2, x_5, x_6\}, \{x_3, x_4\}\}$

$\{2\}^* = \{\{x_1, x_2, x_4, x_5\}, \{x_3, x_6\}\}$

$\{1\}^* \cdot \{2\}^* = \{\{x_1, x_2, x_5\}, \{x_3\}, \{x_4\}, \{x_6\}\}$

That is, for $\{1\}^*$ and $\{2\}^*$, two entities x_1 and x_2 are in the same block of $\{1\}^* \cdot \{2\}^*$ if and only if x_1 and x_2 are in the same block of $\{1\}^*$ and in the same block of $\{2\}^*$. Further, the relaxed covering of *{3}* is *{1, 2}*, because $\{1\}^* \cdot \{2\}^* \leq_r \{3\}^*$, and *{1, 2}* is minimal since $\{1\}^* \leq_r \{3\}^*$ and $\{2\}^* \leq_r \{3\}^*$ are both not true.

TABLE III
DECISION TABLE WITH RELAXED COVERING



	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*L*	*D*	*H*
x_2	*L*	*D*	*H*
x_3	*A*	*C*	*E*
x_4	*A*	*D*	*C*
x_5	*L*	*D*	*H*
x_6	*L*	*C*	-

E. Finding the Set of All Relaxed Coverings

The algorithm R-RICO (Relaxed Rule Induction from Coverings) which is given below can be used to find the set *C* of all relaxed coverings of *R* in *S* (as well as the related rules).

Let *S* be the set of all attributes, and let *R* be the set of all decision attributes. Let *k* be a positive integer. The set of all subsets of the same cardinality *k* of the set *S* is denoted $P_k = \{\{x_{i1}, x_{i2}, \dots , x_{ik}\} \mid x_{i1}, x_{i2}, \dots , x_{ik} \in S\}$.

Algorithm 1: R-RICO

```
begin
    for each attribute x in S do
        compute [x]*;
    compute partition R*
    k:=1
    while k ≤ |S| do
        for each set P in $P_k$ do
            if ($\prod_{x \in P} [x]^* \leq_r R^*$) then
            begin
                find the attribute values from the first block B of P
                and from the first block B’ of R;
                add rule to output file;
            end
        k := k+1;
    end-while
end-algorithm.
```

Note that the condition (iii) for a relaxed covering is not enforced in the R-RICO algorithm. The time complexity of the R-RICO algorithm is exponential to $|S|$, the number of attributes in the data set.

F. RT-RICO Algorithm

The R-RICO algorithm produces rules that are 100% correct. However, unlike decision tree induction, R-RICO produces a more comprehensive rule set. The algorithm can be further modified to satisfy some particular level of uncertainty in the rules (e.g., the rule is ≥ 50% true). That is, rather than just reporting a rule R, we can report the rule as a tuple (R, p) where p is the probability that rule R is true. To accommodate this information in the rules, the definition of attribute dependency inequality must be further modified as in Definition 3.

Definition 3: Relaxed Attribute Dependency Inequality with Threshold

Set R depends on a set P with threshold probability t ($0 < t \leq 1$), and is denoted by $P \rightarrow_{r,t} R$ if and only if $P^* \leq_{r,t} R^*$ and there exists a block B of P^*, and there exists a block B' of R^* such that $(|B \cap B'| / |B|) \geq t$.

It can be observed that, when $t=1$, Definitions 1 and 3 represent the same mathematical relation.

As an example, for the data set of Table IV, let $P = \{1, 2\}$, $R = \{3\}$, and $t = 0.6$. Then we have the following partitions:

$\{1\}^* = \{\{x_1, x_6\}, \{x_2, x_3, x_4, x_5\}\}$

$\{2\}^* = \{\{x_1, x_2, x_3, x_4, x_5, x_6\}\}$

$P^* = \{1,2\}^* = \{1\}^* \cdot \{2\}^* = \{\{x_1, x_6\}, \{x_2, x_3, x_4, x_5\}\}$

$R^* = \{3\}^* = \{\{x_1, x_5\}, \{x_2, x_3, x_4, x_6\}\}$

There exists a block $B = \{x_2, x_3, x_4, x_5\}$ in $\{1, 2\}^*$, and there exists a block $B' = \{x_2, x_3, x_4, x_6\}$ in $\{3\}^*$ such that $(|B \cap B'| / |B|) = |\{x_2, x_3, x_4\}| / |\{x_2, x_3, x_4, x_6\}| = 0.75 \geq 0.6$. Thus, $P^* = \{1, 2\}^* \leq_{r,t} R^* = \{3\}^*$, and $\{3\}$ depends on $\{1, 2\}$ (i.e., $\{1, 2\} \rightarrow_{r,t} \{3\}$), with threshold probability 0.6.

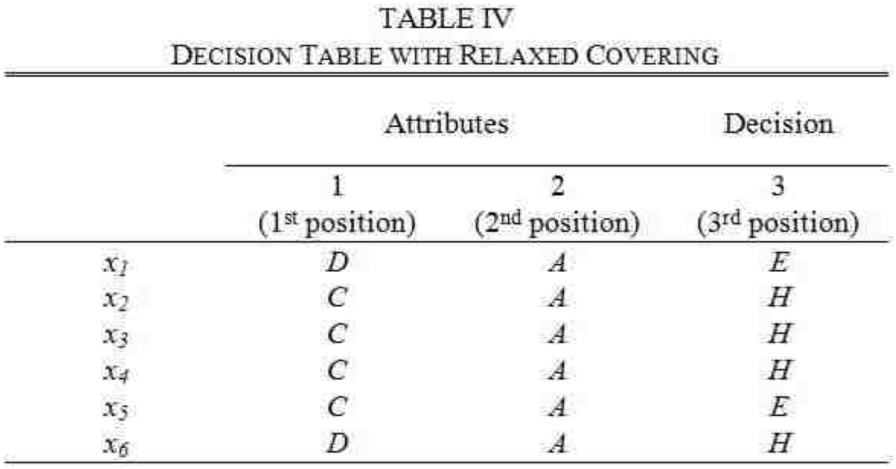


TABLE IV
DECISION TABLE WITH RELAXED COVERING

	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*D*	*A*	*E*
x_2	*C*	*A*	*H*
x_3	*C*	*A*	*H*
x_4	*C*	*A*	*H*
x_5	*C*	*A*	*E*
x_6	*D*	*A*	*H*

We can then find the corresponding values of attributes from entities that are in the region $B \cap B' = \{x_2, x_3, x_4\}$ for the sets $P = \{1, 2\}$ and $R = \{3\}$; namely, the value of attribute 1 is C, the value of attribute 2 is A at $\{x_2, x_3, x_4\}$, and the value of decision 3 is H for entities $\{x_2, x_3, x_4\}$. The rule induced from $\{1, 2\} \rightarrow_{r,t} \{3\}$ is then $(1, C) \wedge (2, A) \rightarrow (3, H)$ with a probability (confidence) of 75%. Another way to look at this is to note that the number of occurrences of $((1,C)(2,A)) = 4$, and the number of occurrences of $((1,C)(2,A) \rightarrow (3, H)) = 3$.

The definition of relaxed coverings must also be modified to incorporate the notion of the threshold probability as in Definition 4.

Definition 4: Relaxed Coverings with Threshold Probability

Let S be a nonempty subset of a set of all attributes, and let R be a nonempty subset of decision attributes, where S and R are disjoint. A subset P of the set S is called a relaxed covering of R in S with threshold probability t $(0 < t \leq 1)$ if and only if $P \rightarrow_{r,t} R$ and P is minimal in S.

Algorithm RT-RICO (Relaxed Threshold Rule Induction From Coverings) finds the set *C* of all relaxed coverings of *R* in *S* (and the related rules), with threshold probability t $(0 < t \leq 1)$, where *S* is the set of all attributes, and *R* is the set of all decisions. The set of all subsets of the same cardinality *k* of the set *S* is denoted $P_k = \{\{x_{i1}, x_{i2}, \ldots, x_{ik}\} \mid x_{i1}, x_{i2}, \ldots, x_{ik} \in S\}$.

Algorithm 2: RT-RICO

```
begin
    for each attribute x in S do
        compute [x]*;
    compute partition R*
    k:=1
    while k ≤ |S| do
        for each set P in $P_k$ do
        if ($\prod_{x \in P}$ [x]* $\leq_{r,t}$ R*) then
        begin
            find values of attributes from the entities that are in the
            region ($B \cap B'$) such that ($|B \cap B'| / |B|$) ≥ t;
            add rule to output file;
        end
        k := k+1
    end-while;
end-algorithm.
```

Note that the condition "*P* is minimal in *S*" of a relaxed covering with threshold probability is not enforced in the RT-RICO algorithm. The reason for not implementing this condition is the same as the reason mentioned in R-RICO algorithm. For our application, to generate rules for protein secondary structure prediction, rules involving more attributes are preferred over rules involving fewer attributes, because they normally generate higher confidence values. Also, we need all the possible attribute position combinations.

The time complexity of the RT-RICO algorithm is again exponential to |*S*|, the number of attributes in the data set. The time complexity is in fact $O(m^2 2^n)$, where *m* is the number of all entities (the number of 5-residue segments), and $n = |S|$ (the number of attributes). 2^n normally dominates the time complexity. For our training data sets, $n = |S| = 5$, and *m* is sufficiently large. Hence, m^2 dominates the time complexity in this case.

As mentioned in Section IV, the rules generated by the RT-RICO algorithm are then compared with the proteins in the test data set to predict the secondary structure elements.

VI. SUMMARY

A novel algorithm, RT-RICO, which generates rules that can be used in predicting protein secondary structure, was presented in this paper. This method performed very well with the training and test data sets used thus far. It should be noted that these preliminary test data sets and training data sets are representative because we selected proteins from different protein families to form them. Specifically, the average prediction accuracy (Q_3 score) of this method was 80.3% (88.7% for “all-α”, 80.2% for “all-β”, 77.0% for “α/β”, and 78.9% for “α+β”).

In the future, we intend to look for ways to further improve the prediction accuracy. In particular, we will analyze how the generated rules actually are used in the prediction process. We then can perform statistical analysis on the specific rules which contribute most (or least) to the prediction results. The statistical analysis may give us ideas on how to improve the prediction score.

At the moment, we favor rules with a 100% confidence value and we measure the choice of secondary structure element by the total support value. We may be able to improve the algorithm in this area by using the rules in different ways. One possible variation of the rule generation process is to use a different threshold value in the RT-RICO algorithm. In this paper, we used a threshold value of 0.9 (90%); hence, we used rules with confidence values from 90% to 100%. If we use a lower threshold value, for example, 0.8 (80%), we should get more rules with higher support values. To effectively use these new rules, we may need to adjust our current prediction algorithm in order to achieve a higher prediction score.

Other interesting questions are how the algorithm will behave if the training data set is a mixture of all four protein type classes, or if we use more proteins in the training data set.

REFERENCES

[1] R. Samudrala, and E. S. Huang, A combined approach for ab initio construction of low resolution protein tertiary structures from sequence. *Pac Symp Biocomput*, 1999, pp. 505-16.

[2] D. T. Jones, Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol*, 1999. 292(2), pp. 195-202.

[3] K. Bryson, L. J. McGuffin, R. L. Marsden, J. J. Ward, J. S. Sodhi, and D. T. Jones, Protein structure prediction servers at University College London. *Nucleic Acids Res*, 2005. 33(Web Server issue), pp. W36-8.

[4] 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. 25(17), pp. 3389-402.

[5] J. A. Cuff, and G. J. Barton, Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins*, 2000. 40(3), pp. 502-11.

[6] B. Rost, Review: protein secondary structure prediction continues to rise. *J Struct Biol*, 2001. 134(2-3), pp. 204-18.

[7] U. Y. Fadime, Y. O¨zlem, and T. Metin, Prediction of secondary structures of proteinsnext term using a two-stage method. *Computers & Chemical Engineering*, 2008. 32(1-2), pp. 78-88

[8] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 2000. 16(5), pp. 412-24.

[9] C. T. Zhang, and R. Zhang, Q_9, a content-balancing accuracy index to evaluate algorithms of protein secondary structure prediction. *Int J Biochem Cell Biol*, 2003. 35(8), pp. 1256-62.

[10] M. Levitt, and C. Chothia, Structural patterns in globular proteins. *Nature*, 1976. 261(5561), pp. 552-8.

[11] Andreeva, D. Howorth, J. M. Chandonia, S. E. Brenner, T. J. Hubbard, C. Chothia C, and A. G. Murzin, Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res*, 2008. 36(Database issue), pp. D419-25.

[12] G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 1995. 247(4), pp. 536-40.

[13] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, The Protein Data Bank. *Nucleic Acids Res*, 2000. 28(1), pp. 235-42.

[14] J. L. Klepeis, and C. A. Floudas, Ab initio prediction of helical segments in polypeptides. *J Comput Chem*, 2002. 23(2), pp. 245-66.

[15] Z. Pawlak, Rough Classification. *Int. J. Man-Machine Studies,* 1984. 20, pp.469-483.

[16] J. W. Grzymala-Busse, *Managing Uncertanity in Expert System.* Boston: Kluwer Academic, 1991, Ch.3.

[17] Y. Y. Koh, V. A. Eyrich, M. A. Marti-Renom, D. Przybylski, M. S. Madhusudhan, N. Eswar, O. Graña, F. Pazos, A. Valencia, A. Sali, and B. Rost, EVA: Evaluation of protein structure prediction servers. *Nucleic Acids Res*, 2003. 31(13), pp. 3311-5.

[18] X. Y. Wang, D. Schroeder, D. Dobbs, and V. Honavar, Data-Driven Discovery of Protein Function Classifiers: Decision Trees Based on MEME Motifs Outperform PROSITE Patterns and Profiles on Peptidase Families. *Proceedings of the 6th Joint Conference on Information Science*, 2002. March 8-13.

[19] X. Wang, J. T. L. Wang, D. Shasha, B. A. Shapiro, I. Rigoutsos, and K. Zhang, Finding Patterns in Three Dimensional Graphs: Algorithms and Applications to Scientific Data Mining, *IEEE Transactions on Knowledge and Data Engineering,* 2003. 14(4), pp. 731-749.

[20] N. E. Davey, D. C. Shields, and R. J. Edwards, SLiMDisc: Short, Linear Motif Discovery, Correcting for Common Evolutionary Descent, *Oxford Journals, Nucleic Acids Research*, 2006. 34(12), pp.3546-3554.

[21] M. Maglia, J. L. Leopold, and V. R. Ghatti, Identifying Character Non-Independence in Phylogenetic Data Using Data Mining Techniques, *Proc. Second Asia-Pacific Bioinformatics Conference Dunedin, New Zealand*, 2004.

[22] J. L. Leopold, A. M. Maglia, M. Thakur, B. Patel, and F. Ercal, Identifying Character Non-Independence in Phylogenetic Data Using Parallelized Rule Induction From Coverings, *Data Mining VIII: Data, Text, and Web Mining and Their Business Applications, WIT Transactions on Information and Communication Technologies,* 2007. 38, pp. 45-54.

[23] J. Han, and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001, pp. 155-157.

PAPER

2. PROTEIN SECONDARY STRUCTURE PREDICTION USING PARALLELIZED RULE INDUCTION FROM COVERINGS

Leong Lee, Cyriac Kandoth, Jennifer L. Leopold, and Ronald L. Frank

Leong Lee, Cyriac Kandoth and Jennifer L. Leopold are affiliated with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: {llkr4, ckhw2, leopoldj}@mst.edu)

Ronald L. Frank is affiliated with the Department of Biological Sciences, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: rfrank@mst.edu)

Abstract**—**Protein 3D structure prediction has always been an important research area in bioinformatics. In particular, the prediction of secondary structure has been a well-studied research topic. Despite the recent breakthrough of combining multiple sequence alignment information and artificial intelligence algorithms to predict protein secondary structure, the Q_3 accuracy of various computational prediction algorithms rarely has exceeded 75%. In a previous paper [1], this research team presented a rule-based method called RT-RICO (Relaxed Threshold Rule Induction from Coverings) to predict protein secondary structure. The average Q_3 accuracy on the sample datasets using RT-RICO was 80.3%, an improvement over comparable computational methods. Although this demonstrated that RT-RICO might be a promising approach for predicting secondary structure, the algorithm's computational complexity and program running time limited its use. Herein a parallelized implementation of a slightly modified RT-RICO approach is presented. This new version of the algorithm facilitated the testing of a much larger dataset of 396 protein domains [2]. Parallelized RT-RICO achieved a Q_3 score of 74.6%, which is higher than the consensus prediction accuracy of 72.9% that was achieved for the same test dataset by a combination of four secondary structure prediction methods [2].

Keywords—data mining, protein secondary structure prediction, parallelization.

I. INTRODUCTION

Prediction of 3D structure of a protein from its amino acid sequence is a very important bioinformatics research goal and has been studied extensively since the 1960s. Protein structure prediction is valuable for drug design, enzyme design, and many other biotechnology applications. Rost [3] suggests that although protein 3D structure prediction from sequence still cannot be achieved fully, in general, research has continuously improved methods for predicting simplified aspects of structure. Particularly in the area of secondary structure prediction, accuracy has surpassed the 70% threshold for all residues of a protein. That breakthrough was achieved by combining multiple sequence alignment information and artificial intelligence algorithms.

It is not an easy task to evaluate the performance of a protein secondary structure prediction method. [2] For example, the use of different datasets for training and testing each algorithm makes it difficult to find an objective comparison of methods. Interestingly, Kabsh and Sanders [4] tested some prediction methods using proteins that had not been used in the development of the algorithms, and found that the reported prediction accuracy of most of those methods decreased by 7 to 27%.

Efforts have been made to develop standard test datasets to accurately evaluate the performance of prediction methods. Cuff and Barton [2] describe the development of a non-redundant test set of 396 protein domains (the CB396 set), where non-redundancy is defined as no two proteins in the set sharing more than 25% sequence identity over a length of more than 80 residues [5]. They used the CB396 set to test four secondary structure prediction methods, PHD [5], DSC [6], PREDATOR [7] and NNSSP [8]. They also combined the four methods by a simple majority-wins method, the CONSENSUS method [2]. The resulting Q_3 scores were 71.9% (PHD), 68.4% (DSC), 68.6% (PREDATOR), 71.4% (NNSSP) and 72.9% for the CONSENSUS method [2].

An interesting secondary structure prediction method described by Fadime, O¨zlem, and Metin [9] uses a two-stage approach. In the first stage, the folding type of a protein is determined. The second stage utilizes data from the Protein Data Bank (PDB) [10] and a probabilistic search algorithm to determine the locations of secondary structure elements. The resulting average accuracy of their prediction score is 74.1%. However, their test dataset is different from the CB396 set.

We previously reported a new method for predicting the secondary structure elements for different folding types [1]. That algorithm, RT-RICO (Relaxed Threshold Rule Induction from Coverings), generates rules for discovering non-independent patterns between protein amino acid sequences and related secondary structure elements. Those rules are then used to predict protein secondary structure. The RT-RICO method performed very well with the training and test datasets used in [1], with a Q_3 accuracy of 80.3%. Although the preliminary test datasets and training datasets used in [1] are representative (i.e., the datasets were made up of proteins selected from different protein families), there was still a need to more extensively test the method. Specifically, to make objective evaluations, different datasets for training and testing needed to be used with RT-RICO.

However, one obstacle to testing RT-RICO with additional datasets was the fact that the algorithm has a time complexity of $O(m^2 2^n)$, where m is the number of all entities (the number of 5-residue segments), and $n = |S|$ (the number of attributes). In practice, n is only 5, while m can be fairly large. Hence, m^2 dominates the time complexity in this case [1]. The largest m value tested was 137,715. When executed on a computer with an Intel Pentium Dual-Core processor, 2 GB of RAM, and Windows XP OS, the total program running time was approximately 14 days.

In order to accommodate a larger dataset (e.g., m value 4,376,003), two new algorithms (Section V, Modified RT-RICO and Parallelization of Modified RT-RICO) were developed. The time complexity of modified RT-RICO is only $O(m \times 2^n)$, although it comes at an acceptable sacrifice of space complexity (i.e., more main memory space is needed as is discussed in Section V). The program was parallelized using an NVIDIA

Tesla C1060 GPU with 4GB of RAM. The 240 cores on this GPU each run at 1.3 GHz. The CPU on the same test machine is a 4-core Intel Core i7-920 with 8GB of RAM. The total program running time improved from days to a few minutes.

The significant improvement of time complexity of the two new algorithms and the subsequent decrease in program running time has enabled us to effectively train and test the RT-RICO method on different available datasets, thereby providing a more objective comparison to other prediction methods. Herein the preliminary results obtained using the improved algorithm are reported.

II. PROBLEM DESCRIPTION

In general, the protein secondary structure prediction problem can be characterized in terms of the following components [11]:

- Input

  Amino acid sequence, $A = a_1, a_2, \ldots a_N$

  Data for comparison, $D = d_1, d_2, \ldots d_N$

  a_i is an element of a set of 20 amino acids, $\{A,R,N\ldots V\}$

  d_i is an element of a set of secondary structures, $\{H,E,C\}$, which represents helix H, sheet E, and coil C.

- Output

  Prediction result: $M = m_1, m_2, \ldots m_N$

  m_i is an element of a set of secondary structures, $\{H,E,C\}$

- 3-Class Prediction [12]

  This is a characterization of the problem as a multi-class prediction problem with 3 classes $\{H,E,C\}$ in which one obtains a 3 × 3 confusion matrix $Z = (z_{ij})$. z_{ij} represents the number of times the input is predicted to be in class j while belonging to class i.

  $Q_{total} = 100 \sum_i Z_{ii} / N$

- Q_3 Score

  Accuracy is computed as $Q_3 = W_{\alpha\alpha} + W_{\beta\beta} + W_{cc}$

  $W_{\alpha\alpha}$ = % of helices correctly predicted

  $W_{\beta\beta}$ = % of sheets correctly predicted

  W_{cc} = % of coils correctly predicted

In other words, a protein secondary structure data sequence *D* is compared to the prediction result sequence *M* to calculate the Q_3 score. It should be noted that in [2], Q_3 is defined a bit differently as:

$Q_3 = \sum_{(i=H,E,C)} predicted_i \,/\, observed_i \times 100$

III. RELATED WORK

In [3], Rost classifies protein secondary structure prediction methods into three generations. The first generation methods depend on single residue statistics to perform prediction. The second generation methods depend on segment statistics. The third generation methods use evolutionary information to predict secondary structure. For example, PHD [5] is a third generation prediction method based on a multiple-level neural network approach. It has been the most accurate method for many years.

One of the best secondary structure predictors is Jones' PSIPRED Protein Structure Prediction Server, which was developed at University College London [13, 14]. PSIPRED uses a two-stage neural network to predict the protein's secondary structure based on position-specific scoring matrices. The matrices are generated by PSI-BLAST (Position-Specific Iterated BLAST) [15]. There are other secondary structure prediction methods that utilize neural network prediction algorithms. For example, Jnet, works by applying multiple sequence alignments alongside profiles such as PSI-BLAST and HMM [16].

Levitt and Chotia proposed to classify proteins as four basic types according to their α-helix and β-sheet content [17]. "All-α" class proteins consist almost entirely (at least 90%) of α-helices. "All-β" class proteins are composed mostly of β-sheets (at least 90%). The "α/β" class proteins have alternating, mainly parallel segments of α-helices and β-sheets. The "α+β" class proteins have a mixture of all-α and all-β regions, mostly in sequential order. Fadime, O¨zlem, and Metin developed a two-stage method to predict secondary structure of proteins [9]. In the first stage of their method, they are able to determine the class of unknown proteins with 100% accuracy. Given a protein sequence, they use a mixed-integer linear program (MILP) approach to decide if the protein sequence belongs to one of the four classes ("all-α", "all-β", "α/β", or "α+β"). In the second stage of their method, they use a probabilistic approach based on their stage one results. They decompose the amino acid sequences of the training set into overlapping sequence groups of three to seven residues. These groups are used to calculate the probability statistics for secondary structure. Specifically, the secondary structure at a particular sequence location is determined by comparing the probabilities that an amino acid residue is a particular secondary structure type based on the statistics.

Their results are impressive. They achieved a 100% accuracy for classifying proteins into one of the four protein type classes ("all-α", "all-β", "α/β", or "α+β"). This greatly simplifies part of the protein secondary structure prediction problem. That is, given a protein amino acid sequence, if it can be determined which one of the four classes this protein belongs to, then other approaches can be applied to predict the secondary structure elements within these four classes. In contrast, our method, RT-RICO, (discussed in detail in [1]) uses a rule-based approach as an alternative way to make the prediction.

A study by Maglia, Leopold and Ghatti [18] implemented a data mining approach based on rule induction from coverings in order to identify non-independence in phylogenetic data. Although rule induction from coverings appeared to be a promising solution for the phylogenetic data non-independence problem, it suffered from exponential computational complexity (which was in part addressed by a parallelized implementation that was tailored for the phylogenetic data by Leopold et al. [19]) as well as the strictness required for the resulting rules (i.e., all rules had to be correct for all instances in the dataset). The restrictive requirement for the rules was addressed in [1], and this allowed the research team to discover meaningful relationships in protein datasets.

IV. RT-RICO APPROACH

RT-RICO (Relaxed Threshold Rule Induction from Coverings) is an implementation of a prediction method given in [1] for solving the protein secondary structure prediction problem. The detailed definitions and algorithms are covered in [1], and hence are not repeated in this paper. In this section, a brief summary of the RT-RICO approach is introduced.

A. RT-RICO Step 1, Data Preparation

As test data, protein names and corresponding folding types of each protein were obtained from the SCOP database [20, 21]. The protein sequences and secondary structure sequences were retrieved from the PDB database [10]. Four databases of proteins (with their amino acid sequences and secondary structure sequences) of different protein types ("all-α", "all-β", "α/β", and "α+β") were built in [1]. Proteins from different protein families were selected to form the training datasets and the test datasets. See Table I for the number of proteins in each training dataset.

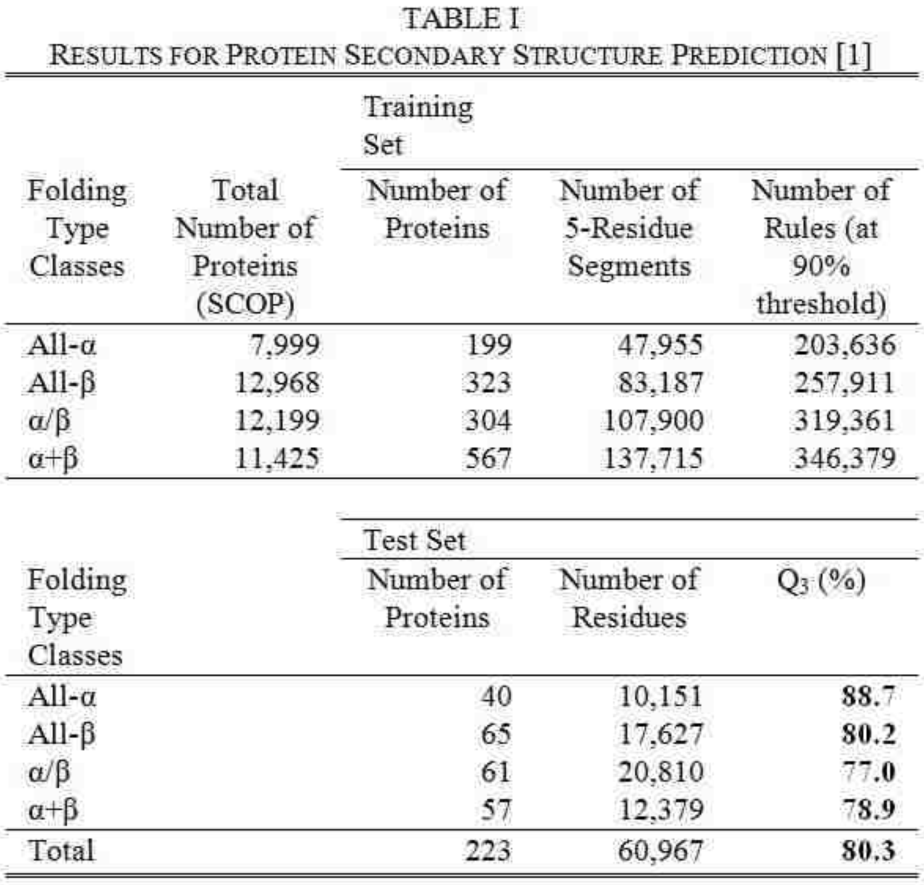


TABLE I
RESULTS FOR PROTEIN SECONDARY STRUCTURE PREDICTION [1]

		Training Set		
Folding Type Classes	Total Number of Proteins (SCOP)	Number of Proteins	Number of 5-Residue Segments	Number of Rules (at 90% threshold)
All-α	7,999	199	47,955	203,636
All-β	12,968	323	83,187	257,911
α/β	12,199	304	107,900	319,361
α+β	11,425	567	137,715	346,379

	Test Set		
Folding Type Classes	Number of Proteins	Number of Residues	Q_3 (%)
All-α	40	10,151	**88.7**
All-β	65	17,627	**80.2**
α/β	61	20,810	**77.0**
α+β	57	12,379	**78.9**
Total	223	60,967	**80.3**

For the first three classes ("all-α", "all-β", and "α/β"), approximately 2.5% of all the available proteins (from SCOP) were chosen as training data. For the "α+β" class, approximately 5% of all the available proteins were chosen as training data. 5% for the last class were chosen mainly because enough 5-residue segments for the "α+β" class were needed. If only 2.5% had been chosen, the number of 5-residue segments for the "α+β" class would be much less than that for the "α/β" class. The PDB IDs for all protein sequences used for training and testing can be found on the following webpage: http://www.leeleong.com/rt-rico/.

The protein secondary structure sequences from PDB are formed by elements of eight states of secondary structure, {H, G, I, E, B, T, S, -}. The eight states were converted to four states to facilitate rule generation as follows:

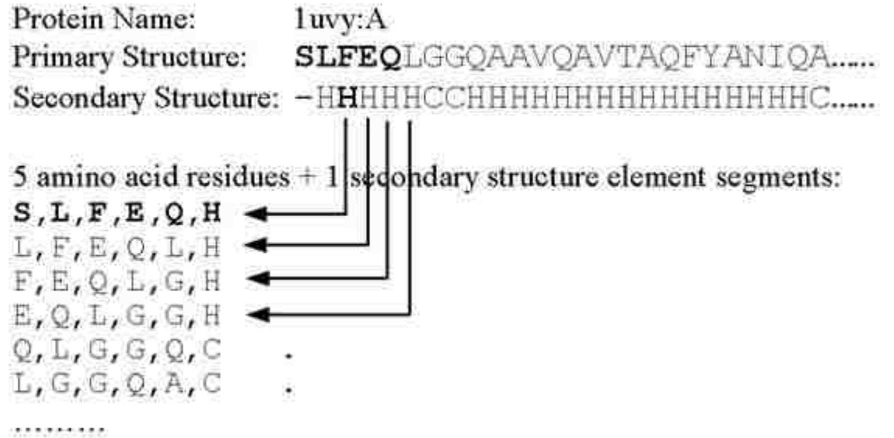
(G, H, I) => Helix H
(E, B) => Sheet E
(T, S) => Coil C
(-) => "-"

Note that rule generation uses a four-state decision attribute. The final Q_3 score calculation uses a three-state decision attribute:

(G, H, I) => Helix H
(E, B) => Sheet E
(Rest) => Coil C

The basis for our approach is to first search segments of amino acid sequences of known protein secondary structures, and then find the rules that relate amino acid residues to secondary structure elements. The generated rules are subsequently used to predict the secondary structure. Klepeis and Floudas showed that the use of overlapping segments of five residues is very effective in predicting the helical segments of proteins [23]. Thus, the overlapping 5-residue segments approach was used to prepare the training data records. As shown in Fig. 1, for each secondary structure element, five "neighboring" amino acid residues were extracted to form a segment of five amino acid residues, plus one secondary structure element. These segments were used as input to the RT-RICO rule generation algorithm to generate rules. The numbers of 5-residue segments generated for the four protein type classes are shown in Table I.

The inputs to RT-RICO are in the form of 6-tuples. The first five elements of a 6-tuple are formed by amino acid residues, {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. The last element of a 6-tuple is formed by one of four secondary structure states {H, E, C, -}. The last element is considered the decision attribute. In other words, the input to RT-RICO Step 2, Rule Generation, are in the form of an $m \times (n+1)$ matrix, where *m* is the number of all entities (the number of 5-residue plus one secondary structure element segments), and *n* = |*S*| (the number of attributes, *n = 5* in this case).


Protein Name: 1uvy:A
Primary Structure: SLFEQLGGQAAVQAVTAQFYANIQA......
Secondary Structure: –HHHHHCCHHHHHHHHHHHHHHHHC......
5 amino acid residues + 1 secondary structure element segments:
S,L,F,E,Q,H
L,F,E,Q,L,H
F,E,Q,L,G,H
E,Q,L,G,G,H
Q,L,G,G,Q,C
L,G,G,Q,A,C
....

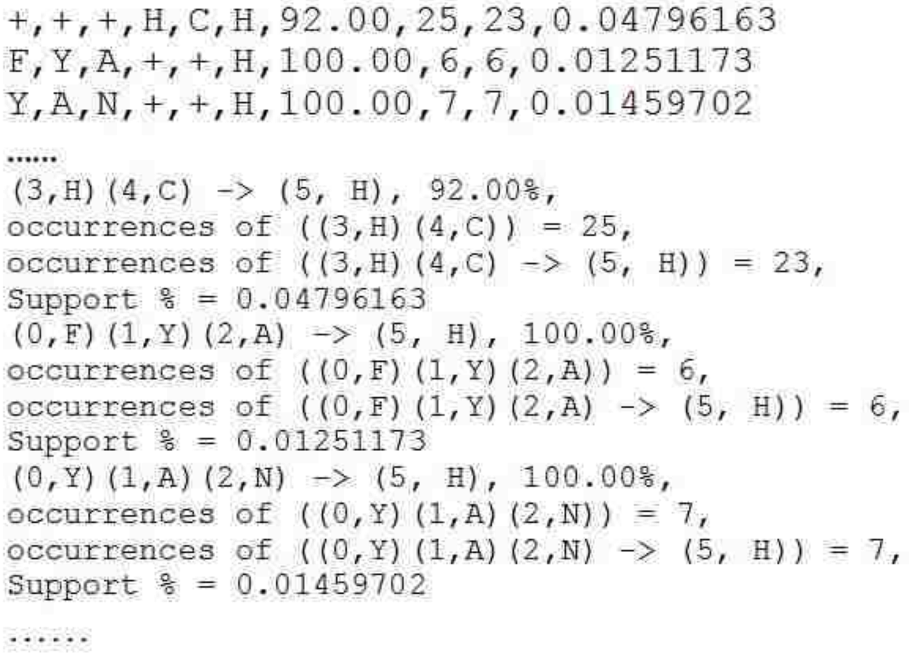
Note: The first and second positions at the beginning of the sequence are represented by 3 residues + 1, and 4 residues + 1 segments, respectively. They form separate training datasets.

Fig. 1. Protein primary structure 5-residue segments and related secondary structure elements representation.

B. RT-RICO Step 2, Rule Generation

RT-RICO generates rules based on the segments in the form of an *m×(n+1)* matrix. Some examples of these rules are shown in Fig. 2 in two separate formats. The first format is intended to be read by the computer programs at the later prediction stage (i.e., the computer rule format). The second format is intended to be read by the user (i.e., the human rule format). The first rule (in human rule format) is interpreted as follows: if the fourth position attribute (or "3" as interpreted by program) is "H", and the fifth position attribute (or "4" as interpreted by program) is "C", then the sixth attribute (decision attribute, or "5" as interpreted by program) is "H" with a confidence of 92% and a support of 0.04796163%. The definitions of confidence and support can be found in [24].

The corresponding first rule (in computer rule format) is interpreted as follows: if the first position attribute is "+" (representing any amino acid element), the second position attribute is "+", the third position attribute is "+", the fourth position attribute is "H", and the fifth position attribute is "C", then the sixth attribute (i.e., the decision attribute) is “H”. The number of occurrences of the fourth position attribute (which is “H”), the fifth position attribute (which is “C”), and the sixth attribute (which is “H”), equals 25 among all inputs to RT-RICO. The number of occurrences of the fourth position attribute (which is “H”) and the fifth position attribute (which is “C”) equals 23 among all inputs to RT-RICO. The support is 0.04796163%.



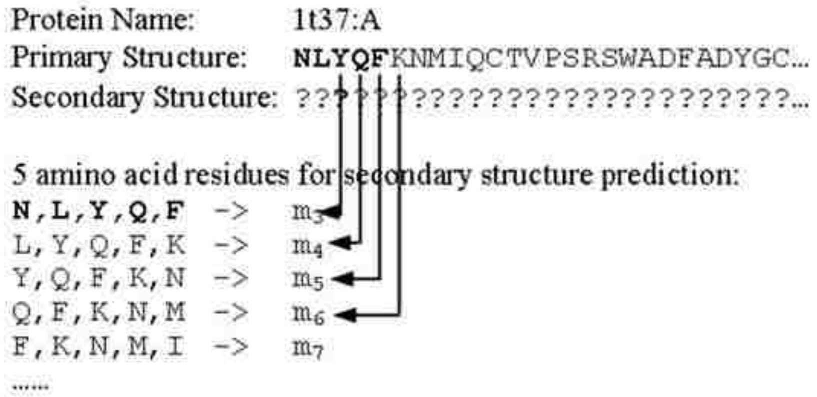
```
+,+,+,H,C,H,92.00,25,23,0.04796163
F,Y,A,+,+,H,100.00,6,6,0.01251173
Y,A,N,+,+,H,100.00,7,7,0.01459702
......
(3,H)(4,C) -> (5, H), 92.00%,
occurrences of ((3,H)(4,C)) = 25,
occurrences of ((3,H)(4,C) -> (5, H)) = 23,
Support % = 0.04796163
(0,F)(1,Y)(2,A) -> (5, H), 100.00%,
occurrences of ((0,F)(1,Y)(2,A)) = 6,
occurrences of ((0,F)(1,Y)(2,A) -> (5, H)) = 6,
Support % = 0.01251173
(0,Y)(1,A)(2,N) -> (5, H), 100.00%,
occurrences of ((0,Y)(1,A)(2,N)) = 7,
occurrences of ((0,Y)(1,A)(2,N) -> (5, H)) = 7,
Support % = 0.01459702
......
```

Fig. 2. Sample rules generated by RT-RICO.

C. RT-RICO Step 3, Prediction

Finally RT-RICO loads protein primary structures from the test dataset, and predicts the secondary structure elements. As shown in Fig. 3, for each secondary structure element prediction position, five “neighboring” amino acid residues are extracted to form a segment of five amino acid residues. Each of these segments is compared with the generated rules. If a segment matches a rule, the support value of the rule is taken into consideration for the prediction of the related secondary structure element. The algorithm first searches for matching rules with 100% confidence value. If no matching rule exists among 100% confidence value rules, the algorithm then searches for other matching rules. The secondary structure element with the highest total support value is selected as the predicted secondary structure element for that specific position. The number of proteins used in the test datasets, and the final Q_3 scores are shown in Table I.

The reported “all-α” proteins have the highest Q_3 score of 88.7%. The “all-β” and “α+β” proteins have Q_3 scores of 80.2% and 78.9%, respectively. The “α/β” proteins have the lowest prediction accuracy of 77.0%.



```
Protein Name:          1t37:A
Primary Structure:     NLYQFKNMIQCTVPSRSWADFADYGC...
Secondary Structure:   ??????????????????????????...

5 amino acid residues for secondary structure prediction:
N,L,Y,Q,F  ->  m3
L,Y,Q,F,K  ->  m4
Y,Q,F,K,N  ->  m5
Q,F,K,N,M  ->  m6
F,K,N,M,I  ->  m7
......
```

Fig. 3. Protein primary structure 5-residue segments and related secondary structure elements prediction. m_i is an element of set {H,E,C,-}. It is then converted to an element of the set {H, E, C}. Note: The first and second positions at the beginning of the sequence are represented (predicted) by 3 residue, and 4 residue segments, respectively. Their related prediction is handled slightly differently.

D. RT-RICO Rule Generation Algorithm

Although the RT-RICO protein secondary structure prediction method consists of the above mentioned three steps, the most computationally intensive part is in the second step - rule generation. Here is a summary of the rule generation algorithm. For detailed definitions used in the algorithm, please refer to [1].

The RT-RICO rule generation algorithm finds the set C of all relaxed coverings of R in S (and the related rules), with threshold probability t $(0 < t \leq 1)$, where S is the set of all attributes, and R is the set of all decisions. The set of all subsets of the same cardinality k of the set S is denoted $P_k = \{\{x_{i1}, x_{i2}, \dots , x_{ik}\} \mid x_{i1}, x_{i2}, \dots , x_{ik} \in S\}$.

```
Algorithm 1: RT-RICO
begin
    for each attribute x in S do
        compute [x]*;
    compute partition R*
    k:=1
    while k ≤ |S| do
        for each set P in $P_k$ do
        if ($\prod_{x \in P}$ [x]* $\leq_{r,t}$ R*) then
        begin
            find values of attributes from the entities that are in the
            region (B ∩ B’) such that (|B ∩ B’| / |B|) ≥  t;
            add rule to output file;
        end
        k := k+1
    end-while;
end-algorithm.
```

The time complexity of the RT-RICO algorithm is exponential with respect to |*S*|, the number of attributes in the dataset. The time complexity is $O(m^2 2^n)$, where *m* is the number of all entities (the number of 5-residue segments), and *n* = |*S*| (the number of attributes). 2^n normally dominates the time complexity. But for our training datasets, *n* is only *5*, while *m* is considerably larger. Hence, m^2 dominates the time complexity in this case.

As mentioned in Section IV(C), the rules generated by the RT-RICO algorithm are then compared with the proteins in the test dataset to predict the secondary structure elements.

E. RT-RICO Running Time Limitations

To more comprehensively evaluate the RT-RICO prediction method, much larger training and test datasets needed to be used to generate rules. In order to improve the RT-RICO time complexity and the program running time, the original rule generation algorithm was modified, and a parallelized strategy was implemented.

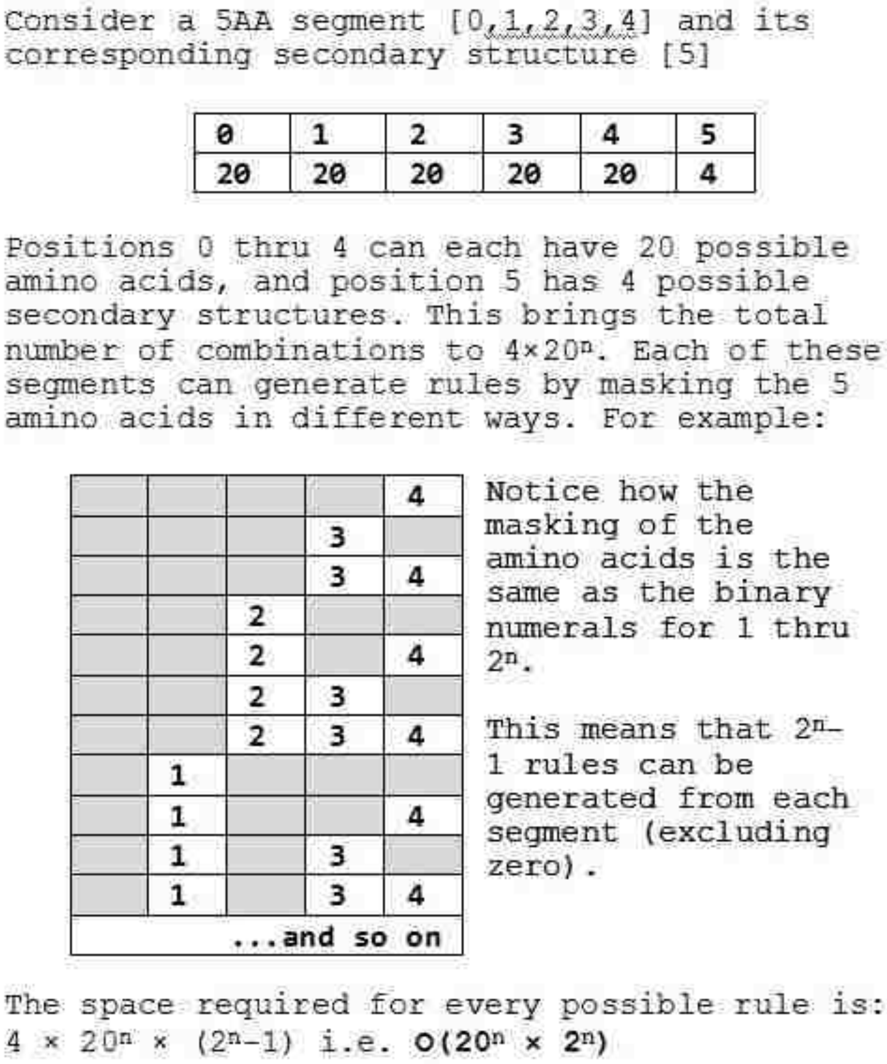
V. PARALLELIZED/MODIFIED RT-RICO ALGORITHMS

The focus of the parallelization of RT-RICO was the rule generation step. It is the most expensive part of the algorithm since it involves generating rules from each segment, counting the frequency of each rule, and finally calculating the confidence and support of each rule. As mentioned earlier, in the sequential implementation of RT-RICO, the complexity of this step is $O(m^2 \times 2^n)$, where m is the number of segments and n the number of amino acid residues in a segment. Usually n is fixed at 5, but m could range from a few thousand to the millions. To reduce the complexity, and hence improve its running time, it was essential to reduce the factor of m in the RT-RICO algorithm.

The m^2 in $O(m^2 \times 2^n)$ is a result of counting the occurrences of each rule. After generating a rule from a segment, the algorithm has to iterate through the list of m segments to count how many times that rule has been seen. This has to be repeated for each of the $m \times 2^n$ rules that can be generated. Hence the complexity is $O(m^2 \times 2^n)$.

But RT-RICO can skip the iteration through the list m times per rule if it simply increments a rule-specific counter every time a rule is generated. The drawback is that there needs to be a counter for every possible rule that can be generated, and this requires an immense amount of main memory. A worst-case calculation of the required space complexity is $O(20^n \times 2^n)$, which translates to approximately 99 Megabytes for 5aa segments, and 163 Gigabytes for 7aa segments. This increases exponentially with an increase in n. The calculation of space complexity is illustrated in Fig. 4.

Despite the exponential space complexity, 5aa segments only require 99 Megabytes of memory. This was further reduced to just 4 Megabytes, by accounting for the duplicate rules that two different segments can generate. For example, the two 5aa segments [S,L,F,E,Q] and [E,L,S,E,Q] can generate the same rule for [+,L,+,E,Q]. The mathematics behind this space optimization is rather complex and is not discussed here, because the 99 Megabytes, or the 4 Megabytes required by the modified algorithm are both trivial amounts on the newer test machine that was used (which has 8192 Megabytes of memory).



Consider a 5AA segment [0,1,2,3,4] and its corresponding secondary structure [5]

0	1	2	3	4	5
20	20	20	20	20	4

Positions 0 thru 4 can each have 20 possible amino acids, and position 5 has 4 possible secondary structures. This brings the total number of combinations to 4×20^n. Each of these segments can generate rules by masking the 5 amino acids in different ways. For example:

				4
			3	
			3	4
		2		
		2		4
		2	3	
		2	3	4
	1			
	1			4
	1		3	
	1		3	4
...and so on				

Notice how the masking of the amino acids is the same as the binary numerals for 1 thru 2^n.

This means that 2^n-1 rules can be generated from each segment (excluding zero).

The space required for every possible rule is:
$4\times20^n\times(2^n-1)$ i.e. **$O(20^n\times2^n)$**

Fig. 4. The number of all possible rules from 5aa segments.

A. Modified Algorithm for Rule Generation

In essence, the modified RT-RICO algorithm compromises on space complexity for the sake of reducing time complexity. Algorithm 2 describes this modification is more detail.

Algorithm 2: Modified RT-RICO

```
begin
    Allocate counters for every possible rule (initialize to 0)
    for each segment
        for each $2^n$-1 rules from this segment
            Calculate the memory location of the counter
            corresponding to this rule, and increment it by 1
        end-for
    end-for
    Read each counter and calculate the confidence and support for those rules
    that pass the relaxed threshold
end-algorithm.
```

The complexity of this algorithm is just $O(m\times 2^n)$ because the algorithm does not need to count the reoccurrence of each rule. The generated rules simply increment a counter whenever they are generated. There is an additional amount of time required to calculate the memory location of the counter that corresponds to a rule. However, this is negligible, and as a constant, it does not affect the overall complexity of the algorithm.

B. Parallelization of Rule Generation

The modified RT-RICO rule generation algorithm places no restrictions on the order in which rules are generated. So parallelizing the algorithm involves a straightforward distribution of the input data among processing units. Each processing unit calculates the memory location of the counter corresponding to the rule that it generates from a given segment, and increments that counter. These operations can be performed in parallel by any number of concurrent processing units. However, for performance reasons (e.g., to minimize potentially conflicting concurrent updates of shared memory locations), the number of concurrent processing units is kept under a predetermined threshold.

C. Massively Parallel Computation using GPUs

Compute Unified Device Architecture (CUDA) is a programming interface for developing general purpose applications on Graphics Processing Units (GPUs). GPUs are conventionally used for graphics acceleration, which typically involves repeatedly performing the same computational operation on multiple input data, also known as SIMD operations (single instruction multiple data). Because of the constraints placed on SIMD operations, GPU hardware is designed with features such as massively parallel processing and pipelining to accelerate the execution of these operations. With CUDA, GPUs can be directly programmed using the C programming language to process any kind of general purpose operation, which would normally be tasked to CPUs. However, because the GPU hardware remains the same, they are still ideally suited for SIMD operations, and more complex operations are likely to run faster sequentially on a CPU.

The modified RT-RICO rule generation algorithm is an ideal SIMD operation. The calculation of the memory location of the counter that corresponds to a rule extracted from a segment, is performed over and over again for all the given segments in the input file. This SIMD operation was parallelized using an NVIDIA Tesla C1060 GPU with 4GB of RAM. The 240 cores on this GPU each run at 1.3 GHz. The CPU on the same test machine was a 4-core Intel Core i7-920 with 8GB of RAM. The total program running time was approximately 3 minutes and 33 seconds for rule generation of the dataset in Table II, which is much larger than the dataset of Table I.

VI. RESULTS

A standard test dataset of 396 protein domains (the CB396 set developed by Cuff and Barton [2]) was used to evaluate the performance of the new parallelized, modified RT-RICO rule generation algorithm, and also the overall RT-RICO prediction performance. See Table II for the number of proteins in each training dataset, and the performance of RT-RICO prediction method on CB396 test dataset.

The CB396 dataset is a specially developed non-redundant test dataset created with the objective of comparing different protein secondary structure prediction methods. In [2], the CB396 set was applied to four secondary structure prediction methods and a CONSENSUS method. Respectively, the Q_3 scores were 71.9% (PHD [5]), 68.4% (DSC

[6]), 68.6% (PREDATOR [7]), 71.4% (NNSSP [8]) and 72.9% for the CONSENSUS method (which combined the above four methods) [2]. The parallelization of RT-RICO enabled us to test our approach using the CB396 test dataset.

The final Q_3 scores of RT-RICO prediction of CB396 test dataset are shown in Table II. The "all-α" protein domains have the highest Q_3 score of 82.6%. The "all-β" and "α/β" protein domains have Q_3 scores of 77.4% and 72.9%, respectively. The "α+β" and "Others" protein domains have the prediction accuracy of 71.3% and 69.5%. On average, RT-RICO has a Q_3 score of 74.6%, which is higher than the Q_3 score generated by other methods using the same test dataset (as reported in [2]).

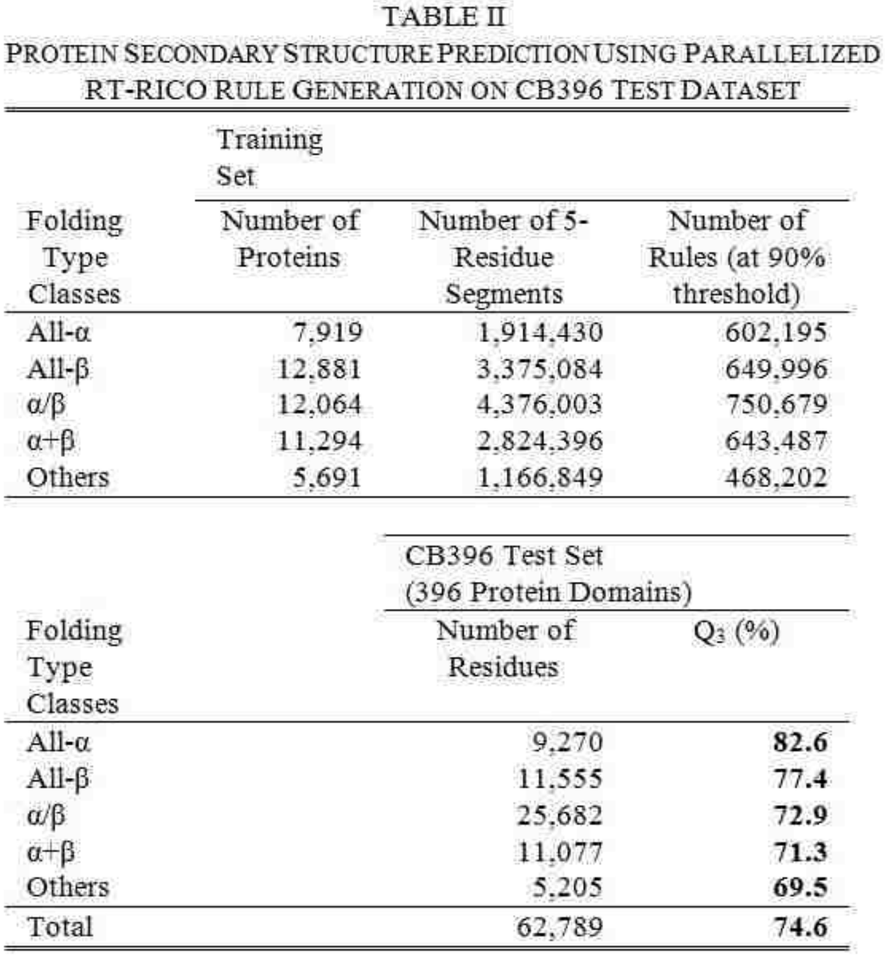


TABLE II
PROTEIN SECONDARY STRUCTURE PREDICTION USING PARALLELIZED RT-RICO RULE GENERATION ON CB396 TEST DATASET

	Training Set		
Folding Type Classes	Number of Proteins	Number of 5-Residue Segments	Number of Rules (at 90% threshold)
All-α	7,919	1,914,430	602,195
All-β	12,881	3,375,084	649,996
α/β	12,064	4,376,003	750,679
α+β	11,294	2,824,396	643,487
Others	5,691	1,166,849	468,202

	CB396 Test Set (396 Protein Domains)	
Folding Type Classes	Number of Residues	Q_3 (%)
All-α	9,270	**82.6**
All-β	11,555	**77.4**
α/β	25,682	**72.9**
α+β	11,077	**71.3**
Others	5,205	**69.5**
Total	62,789	**74.6**

VII. CONCLUSION

Despite the large amount of available protein data, applying the originally developed RT-RICO prediction method [1] to predict protein secondary structure was difficult. The lengthy program running time primarily was the result of the $O(m^2 2^n)$ time complexity of the rule generation step. Therefore, two new algorithms were developed (Section V, Modified RT-RICO and Parallelization of Modified RT-RICO). The time complexity of modified RT-RICO is only $O(m \times 2^n)$, although it comes at an acceptable sacrifice of space complexity. The resulting faster running time of the program facilitated the use of the CB396 test dataset to test the RT-RICO prediction method. For that dataset the average Q_3 accuracy of the RT-RICO predictions was 74.6%, which is higher than the Q_3 scores generated by other prediction methods using the same dataset (as reported in [2]). In the future, the research team plans to use other available standard test datasets to further objectively evaluate the performance of this new, promising prediction method, as well as to continue to look for ways to improve the accuracy of the predictions.

REFERENCES

[1] L. Lee, J. L. Leopold, R. L. Frank and A. M. Maglia, “Protein Secondary Structure Prediction Using Rule Induction from Coverings,” *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology 2009 (part of IEEE Symposium Series on Computational Intelligence 2009)*, Nashville, Tennessee, USA, pp. 79-86.

[2] J. A. Cuff, and G. Barton, “Evaluation and improvement of multiple sequence methods for protein secondary structure prediction,” *Proteins,* 34, pp. 508–519, 1999.

[3] B. Rost, “Rising accuracy of protein secondary structure prediction”, D. Chasman, Ed., *Protein structure determination, analysis, and modeling for drug discovery*, New York: Dekker, 2003, pp. 207–249.

[4] W. Kabsh and C. Sander, “How good are predictions of protein secondary structure?”, *FEBS Letters*, 155, pp. 179-182, 1983.

[5] B. Rost and C. Sander, “Prediction of protein secondary structure at better than 70% accuracy,” *J. Mol. Biol.*, 232, pp. 584-599, 1993.

[6] R. D. King and M. J. E. Sternberg, “Identification and application of the concepts important for accurate and reliable protein secondary structure prediction,” *Protein Sci*, 1996, 5, pp. 2298–2310.

[7] D. Frishman and P. Argos, “Seventy-five percent accuracy in protein secondary structure prediction,” *Proteins,* 1997, 27, pp. 329–335.

[8] A. A. Salamov and V. V. Solovyev, “Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments,” *J Mol Biol*, 1995, 247, pp. 11–15.

[9] U. Y. Fadime, Y. O¨zlem, and T. Metin, “Prediction of secondary structures of proteinsnext term using a two-stage method,” *Computers & Chemical Engineering*, 2008. 32(1-2), pp. 78-88.

[10] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The Protein Data Bank,” *Nucleic Acids Res*, 2000, 28(1), pp. 235-42.

[11] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: an overview,” *Bioinformatics*, 2000. 16(5), pp. 412-24.

[12] C. T. Zhang, and R. Zhang, “Q_9, a content-balancing accuracy index to evaluate algorithms of protein secondary structure prediction,” *Int J Biochem Cell Biol*, 2003. 35(8), pp. 1256-62.

[13] D. T. Jones, “Protein secondary structure prediction based on position-specific scoring matrices,” *J Mol Biol,* 1999, 292(2), pp. 195-202.

[14] K. Bryson, L. J. McGuffin, R. L. Marsden, J. J. Ward, J. S. Sodhi, and D. T. Jones, “Protein structure prediction servers at University College London”, *Nucleic Acids Res*, 2005, 33(Web Server issue), pp. W36-8.

[15] 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, 25(17), pp. 3389-402.

[16] J. A. Cuff, and G. J. Barton, “Application of multiple sequence alignment profiles to improve protein secondary structure prediction,” *Proteins,* 2000, 40(3), pp. 502-11.

[17] M. Levitt, and C. Chothia, “Structural patterns in globular proteins,” *Nature*, 1976, 261(5561), pp. 552-8.

[18] A. M. Maglia, J. L. Leopold, and V. R. Ghatti, “Identifying Character Non-Independence in Phylogenetic Data Using Data Mining Techniques,” *Proc. Second Asia-Pacific Bioinformatics Conference Dunedin, New Zealand*, 2004.

[19] J. L. Leopold, A. M. Maglia, M. Thakur, B. Patel, and F. Ercal, “Identifying Character Non-Independence in Phylogenetic Data Using Parallelized Rule Induction From Coverings,” *Data Mining VIII: Data, Text, and Web Mining and Their Business Applications, WIT Transactions on Information and Communication Technologies,* 2007, 38, pp. 45-54.

[20] A. Andreeva, D. Howorth, J. M. Chandonia, S. E. Brenner, T. J. Hubbard, C. Chothia C, and A. G. Murzin, “Data growth and its impact on the SCOP database: new developments,” *Nucleic Acids Res*, 2008, 36(Database issue), pp. D419-25.

[21] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, “SCOP: a structural classification of proteins database for the investigation of sequences and structures,” *J Mol Biol*, 1995, 247(4), pp. 536-40.

[22] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The Protein Data Bank,” *Nucleic Acids Res*, 2000, 28(1), pp. 235-42.

[23] J. L. Klepeis, and C. A. Floudas, “Ab initio prediction of helical segments in polypeptides,” *J Comput Chem*, 2002, 23(2), pp. 245-66.

[24] J. Han, and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001, pp. 155-157.

PAPER

3. PROTEIN SECONDARY STRUCTURE PREDICTION USING RT-RICO: A RULE-BASED APPROACH

Leong Lee[*,1], Jennifer L. Leopold[1], Cyriac Kandoth[1] and Ronald L. Frank[2]

[1]*Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA*

[2]*Department of Biological Sciences, Missouri University of Science and Technology, Rolla, MO, USA*

Abstract

Protein structure prediction has always been an important research area in biochemistry. In particular, the prediction of protein secondary structure has been a well-studied research topic. The experimental methods currently used to determine protein structure are accurate, yet costly both in terms of equipment and time. Despite the recent breakthrough of combining multiple sequence alignment information and artificial intelligence algorithms to predict protein secondary structure, the Q_3 accuracy of various computational prediction methods rarely has exceeded 75%. In this paper, a newly developed rule-based data-mining approach called RT-RICO (Relaxed Threshold Rule Induction from Coverings) is presented. This method identifies dependencies between amino acids in a protein sequence and generates rules that can be used to predict secondary structure. RT-RICO achieved a Q_3 score of 81.75% on the standard test dataset RS126 and a Q_3 score of 79.19% on the standard test dataset CB396, an improvement over comparable computational methods.

Keywords:

Data mining; Protein secondary structure prediction; Parallelization.

1. Introduction

Prediction of 3D structure of a protein from its amino acid sequence is a very important research goal in biochemistry and bioinformatics, and has been studied extensively since the 1960s. Protein structure prediction is valuable for drug design, enzyme design, and many other biotechnology applications. Rost [1] suggests that although protein 3D structure prediction from sequence still cannot be achieved fully, in general, research has continuously improved methods for predicting simplified aspects of structure. Particularly in the area of secondary structure prediction, accuracy has surpassed the 70% threshold for all residues of a protein. That breakthrough was achieved by combining multiple sequence alignment information and artificial intelligence algorithms.

It is not an easy task to evaluate the performance of a protein secondary structure prediction method. For example, the use of different datasets for training and testing each algorithm makes it difficult to find an objective comparison of methods [2]. Interestingly, Kabsh and Sanders [3] tested prediction methods using proteins that had not been used in the development of the algorithms and found that the reported prediction accuracy of most of those methods decreased by more than 7%. One method's prediction accuracy decreased by as much as 27%. Rost [1] stated that "there is no value in comparing methods evaluated on different datasets."

Efforts have been made to develop standard test datasets to accurately evaluate the performance of prediction methods. Rost and Sander [4] selected a list of 126 protein domains (the RS126 set) that now constitutes a comparative standard.

Cuff and Barton [2] described the development of a non-redundant test set of 396 protein domains (the CB396 set) where non-redundancy is the case; no two proteins in the set share more than 25% sequence identity over a length of more than 80 residues [4]. They used the CB396 set to test four secondary structure prediction methods: PHD [4], DSC [5], PREDATOR [6] and NNSSP [7]. They also combined the four methods by a simple majority-wins method, the CONSENSUS method [2]. The resulting Q_3 scores for the CB396 set were 71.9% (PHD), 68.4% (DSC), 68.6% (PREDATOR), 71.4% (NNSSP) and 72.9% for the CONSENSUS method.

In the same research study, Cuff and Barton [2] also tested the RS126 set in which the Q_3 scores were 73.5% (PHD), 71.1% (DSC), 70.3% (PREDATOR), 72.7% (NNSSP) and 74.8% for the CONSENSUS method.

An interesting secondary structure prediction method described by Fadime, O¨zlem and Metin [8] uses a two-stage approach. In the first stage, the folding type of a protein is determined. The second stage utilizes data from the Protein Data Bank (PDB) [9] and a probabilistic search algorithm to determine the locations of secondary structure elements. The resulting average accuracy of their prediction score is 74.1%. However, the test dataset was not RS126 or CB396.

In this paper, we present a new method for predicting the secondary structure elements for different folding types. The algorithm, RT-RICO (Relaxed Threshold Rule Induction from Coverings), generates rules for discovering dependencies between protein amino acid sequences and related secondary structure elements. These rules are then used to predict protein secondary structure. The RT-RICO method performed better than previously reported methods, with a Q_3 accuracy of 81.75% on the RS126 set and 79.19% on the CB396 set.

The RT-RICO approach and the main RT-RICO rule generation algorithm are discussed in Sections 3 and 4. A parallelized version of this algorithm is presented in Section 5, and detailed results of this method are presented in Section 6.

2. Related Work

Rost [1] classifies protein secondary structure prediction methods into three generations. The first generation methods depend on single residue statistics to perform prediction. The second generation methods depend on segment statistics. The third generation methods use evolutionary information to predict secondary structure. For example, PHD [4] is a third generation prediction method based on a multiple-level neural network approach. It has been the most accurate method for many years.

One of the best secondary structure predictors is PSIPRED Protein Structure Prediction Server [10], which was developed at University College London [10, 11]. PSIPRED uses a two-stage neural network to predict the protein’s secondary structure based on position-specific scoring matrices. The matrices are generated by PSI-BLAST

(Position-Specific Iterated BLAST) [12]. The PSIPRED's Q_3 score based on a set of 187 unique folds is between 76.5% and 78.3% [10]. There are other secondary structure prediction methods that utilize neural network prediction algorithms. For example, Jnet works by applying multiple sequence alignments alongside profiles such as PSI-BLAST and HMM [13].

Random errors in the DNA sequence lead to a different translation of protein sequences. These 'errors' are the basis for evolution [1]. Due to the fact that mutations resulting in a structural change are not likely to survive, Rost states that the evolutionary pressure to conserve structure and function has led to a record of the unlikely event: structure is more conserved than sequence [1]. Many third generation methods capitalize on this event to improve prediction accuracy. In PHD [4], Rost and Sander use multiple sequence alignments rather than single sequences as input to a neural network. At the training stage, a database of protein families aligned to proteins of known structure is used. At the prediction stage, the database of sequences is scanned for all homologues of the protein to be predicted, and the family profile of amino acid frequencies at each alignment position is fed into the network [14]. PSIPRED take advantage of the same concept, but uses a slightly different approach, via matrices generated by PSI-BLAST [10].

These artificial neural network methods are revolutionary in the sense that they employ the homologues of proteins for training and prediction. It is considered that a neural network is like a "black box"; it is difficult to formulate an algorithm from a neural network. A trained network may succeed in solving a problem, but it is hard to understand how it works. As a result, we are inspired to utilize a different approach, a rule-based prediction method. This approach still makes use of the fundamental principle that structure is more conserved than sequence. We establish rules between each known secondary structure element and its "neighboring" amino acid residues. These rules are used to perform predictions. Due to the different approaches, it is difficult to directly compare prediction results between this method and other methods. Neural network methods normally employ rigorous cross-validation testing techniques. The final Q_3 scores comparison should be used as a general guide, not a strict percentile comparison.

Recently, there is a trend using the support vector machine (SVM) to predict protein secondary structures. Hu, Pan, Harrison and Tai [15] achieved a Q_3 accuracy of 78.8% on the RS126 dataset using a SVM approach. Kim and Park [16] developed the SVMpsi method that resulted in Q_3 scores of 76.1% on the RS126 dataset and 78.5% on their KP480 dataset. Nguyen and Rajapakse [17] proposed a two-stage multi-class SVM approach utilizing position-specific scoring matrices generated by PSI-BLAST. Their Q_3 scores were 78.0% on the RS126 dataset and 76.3% on the CB396 dataset.

Levitt and Chothia [18] proposed to classify proteins as four basic types according to their α-helix and β-sheet content. "All-α" class proteins consist almost entirely (at least 90%) of α-helices. "All-β" class proteins are composed mostly of β-sheets (at least 90%). The "α/β" class proteins have alternating, mainly parallel segments of α-helices and β-sheets. The "α+β" class proteins have a mixture of all-α and all-β regions, mostly in sequential order. The first stage of the two stage method developed by Fadime, O¨zlem and Metin [8] is able to determine the class of unknown proteins with 100% accuracy. Given a protein sequence, they use a mixed-integer linear program (MILP) approach to decide if the protein sequence belongs to one of the four classes ("all-α", "all-β", "α/β", or "α+β"). In the second stage they use a probabilistic approach based on their stage one results. The amino acid sequences of the training set are distributed into overlapping sequence groups of three to seven residues. These groups are used to calculate the probability statistics for secondary structure. Specifically, the secondary structure at a particular sequence location is determined by comparing the probabilities that an amino acid residue is a particular secondary structure type based on the statistics.

Their results are impressive. They achieved a 100% accuracy for classifying proteins into one of the four protein type classes ("all-α", "all-β", "α/β", or "α+β"). This greatly simplifies part of the protein secondary structure prediction problem. That is, given a protein amino acid sequence, if it can be determined which one of the four classes this protein belongs to, then other approaches can be applied to predict the secondary structure elements within these four classes. In contrast, the RT-RICO method uses a rule-based approach as an alternative way to make the prediction.

A study by Maglia, Leopold, and Ghatti [19] implemented a data mining approach based on rule induction from coverings in order to identify non-independence in phylogenetic data. Although rule induction from coverings appeared to be a promising solution for the phylogenetic data non-independence problem, it suffered from exponential computational complexity (which was in part addressed by a parallelized implementation that was tailored for the phylogenetic data [20]) as well as the strictness required for the resulting rules (i.e., all rules had to be correct for all instances in the dataset). The restrictive requirement for the rules is addressed in Section 3, and this allowed the research team to discover meaningful rules in another problem domain, protein datasets.

Kabsch and Sander developed a set of simple and physically motivated criteria for secondary structure, programmed as a pattern-recognition process of hydrogen-bonded and geometrical features extracted from x-ray coordinates [21]. This DSSP (Define Secondary Structure of Proteins) algorithm is the standard method for assigning secondary structure to the primary structure (amino acids) of a protein. Depending on the pattern of hydrogen bonds, DSSP recognizes eight types or states of secondary structure. The 3-helix (3/10 helix), alpha helix, and 5 helix (pi helix) are symbolized as G, H and I, respectively. DSSP recognizes two types of hydrogen-bond pairs in beta sheet structures, the parallel and antiparallel bridge. Residue in isolated beta-bridge is symbolized by B, whereas E represents an extended strand, and participates in a beta ladder. The remaining types are T for hydrogen bonded turn, and S for bend. There is also blank or "-" meaning "loop" or "other." These eight types are usually grouped into three classes: helix (G, H, and I), strand/sheet (E and B) and loop/coil (all others).

3. RT-RICO Approach

3.1. Problem Description

In general, the protein secondary structure prediction problem can be characterized in terms of the following components [22]:

- Input

  Amino acid sequence, $A = a_1, a_2, \ldots a_N$

  Data for comparison, $D = d_1, d_2, \ldots d_N$

  a_i is an element of a set of 20 amino acids, $\{A,R,N \ldots V\}$

d_i is an element of a set of secondary structures, $\{H,E,C\}$, which represents helix H, sheet E, and coil C.

- Output

  Prediction result: $M = m_1, m_2, \ldots m_N$

  m_i is an element of a set of secondary structures, $\{H,E,C\}$

- 3-Class Prediction [23]

  This is a characterization of the problem as a multi-class prediction problem with 3 classes $\{H,E,C\}$ in which one obtains a 3 × 3 confusion matrix $Z = (z_{ij})$. z_{ij} represents the number of times the input is predicted to be in class j while belonging to class i.

  $Q_{total} = 100 \sum_i Z_{ii} / N$

- Q_3 Score

  Accuracy is computed as $Q_3 = W_{\alpha\alpha} + W_{\beta\beta} + W_{cc}$

  $W_{\alpha\alpha}$ = % of helices correctly predicted

  $W_{\beta\beta}$ = % of sheets correctly predicted

  W_{cc} = % of coils correctly predicted

  In other words, a protein secondary structure data sequence D is compared to the prediction result sequence M to calculate the Q_3 score.

3.2. RT-RICO Step 1, Data Preparation

RT-RICO (Relaxed Threshold Rule Induction from Coverings) is the implementation of a prediction method for solving the protein secondary structure prediction problem. First, all protein names and corresponding folding types of each protein are retrieved from the SCOP database [24, 25]. All available corresponding protein sequences and secondary structure sequences are retrieved from the PDB database [9]. Five databases of protein domains (with their amino acid sequences and secondary structure sequences) of different protein domain types (“all-α”, “all-β”, “α/β”, “α+β” and “others”) are built. Proteins from the test datasets (RS126 or CB396) are first removed from these databases, so that they will be excluded from the possible training datasets. Protein domains from different protein families are selected to form the training datasets. See Table 1 for the number of protein domains in each training dataset on the RS126 test dataset.

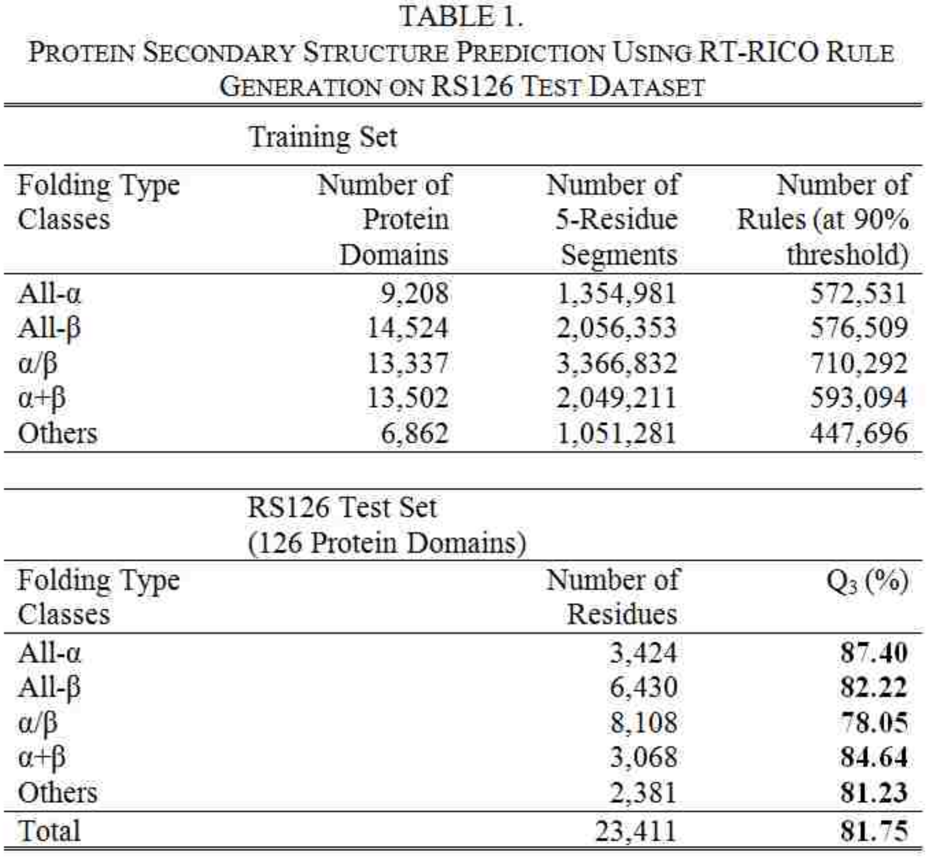


TABLE 1.
PROTEIN SECONDARY STRUCTURE PREDICTION USING RT-RICO RULE GENERATION ON RS126 TEST DATASET

	Training Set		
Folding Type Classes	Number of Protein Domains	Number of 5-Residue Segments	Number of Rules (at 90% threshold)
All-α	9,208	1,354,981	572,531
All-β	14,524	2,056,353	576,509
α/β	13,337	3,366,832	710,292
α+β	13,502	2,049,211	593,094
Others	6,862	1,051,281	447,696

	RS126 Test Set (126 Protein Domains)	
Folding Type Classes	Number of Residues	Q_3 (%)
All-α	3,424	**87.40**
All-β	6,430	**82.22**
α/β	8,108	**78.05**
α+β	3,068	**84.64**
Others	2,381	**81.23**
Total	23,411	**81.75**

The protein secondary structure sequences from PDB are formed by elements of eight states of secondary structure, {H, G, I, E, B, T, S, -}. The eight states are converted to four states to facilitate rule generation as follows:

(G, H, I) => Helix H

(E, B) => Sheet E

(T, S) => Coil C

(-) => “-”

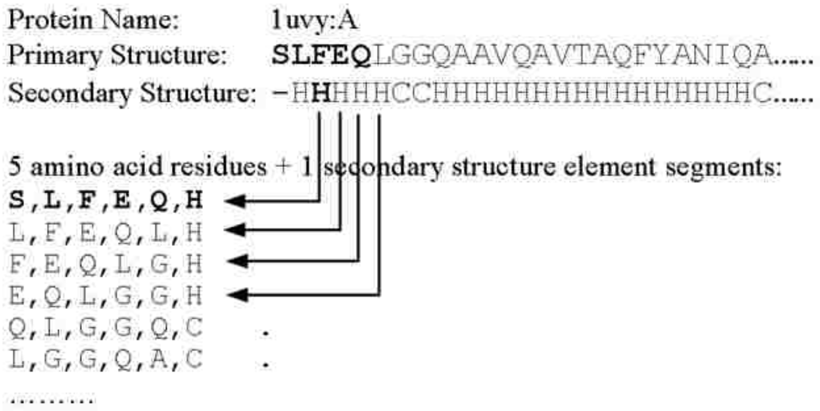
Note that rule generation uses a four-state decision attribute. The final Q_3 score calculation uses a three-state decision attribute:

(G, H, I) => Helix H

(E, B) => Sheet E

(Rest) => Coil C

The basis for the RT-RICO approach is to first search segments of amino acid sequences of known protein secondary structures, and then find the rules that relate amino acid residues to secondary structure elements. The generated rules are subsequently used to predict the secondary structure. Klepeis and Floudas [26] showed that the use of overlapping segments of five residues is very effective in predicting the helical segments of proteins. Thus, the overlapping 5-residue segments approach was used to prepare the training data records. As shown in Fig. (**1**), for each secondary structure element, five “neighboring” amino acid residues are extracted to form a segment of five amino acid residues, plus one secondary structure element. These segments are used as input to the RT-RICO rule generation algorithm (Section 3.3, with more detail in Section 4) to generate rules. The numbers of 5-residue segments generated for the five protein type classes are shown in Table 1.


Protein Name: 1uvy:A
Primary Structure: SLFEQLGGQAAVQAVTAQFYANIQA......
Secondary Structure: −HHHHHCCHHHHHHHHHHHHHHHHC......
5 amino acid residues + 1 secondary structure element segments:
S,L,F,E,Q,H
L,F,E,Q,L,H
F,E,Q,L,G,H
E,Q,L,G,G,H
Q,L,G,G,Q,C
L,G,G,Q,A,C
....

Note: The first and second positions at the beginning of the sequences are represented by 3 residues + 1, and 4 residues + 1 segments, respectively. They form separate training datasets.

Fig. (1). Protein primary structure 5-residue segments and related secondary structure elements representation.

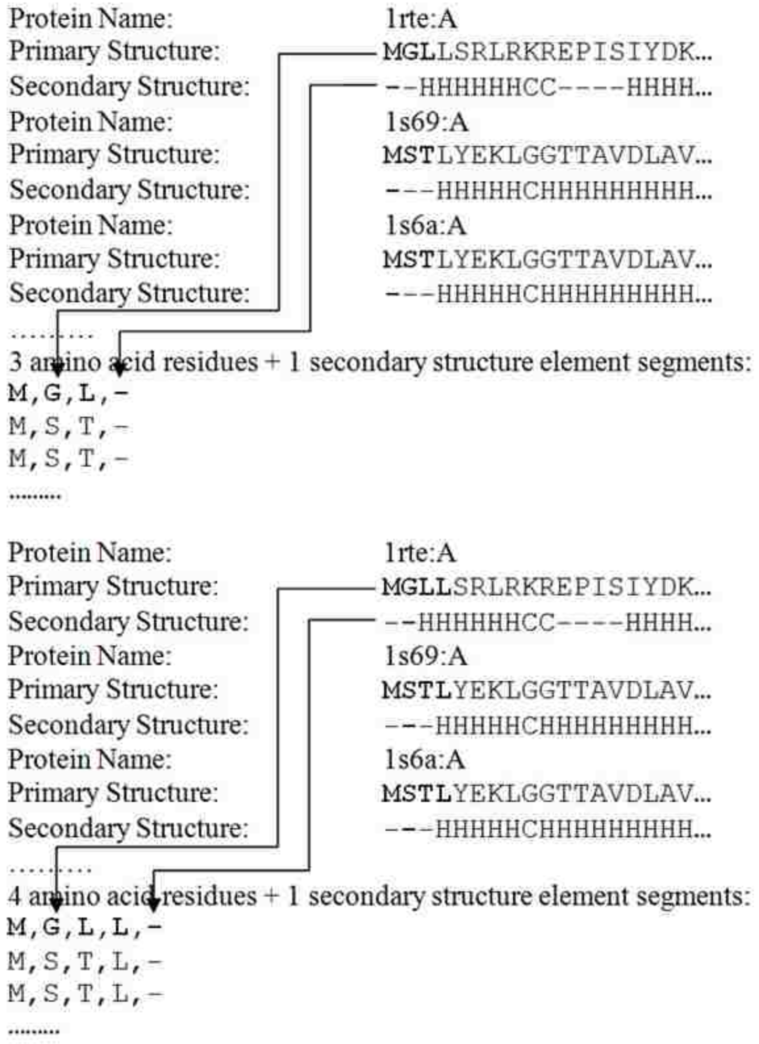
Although we use 5-residue segments, there is no evidence that five is the best segment length for this algorithm. PSIPRED uses a window of 15 amino acid residues for the neural network design [10]. Most previous methods combine multiple sequence alignment information and machine learning techniques. The purpose is to find the highly-correlated patterns from the training databases. A challenging future research problem remaining for RT-RICO is how to choose the best residue segment length, hence extracting correct and concise rules.

The main inputs to the RT-RICO rule generation algorithm are in the form of 6-tuples. The first five elements of a 6-tuple are formed by amino acid residues, {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. The last element of a 6-tuple is formed by one of four secondary structure states {H, E, C, -}. The last element is considered the decision attribute. In other words, the input to RT-RICO Step 2, Rule Generation, is in the form of an $m\times(n+1)$ matrix, where m is the number of all entities (the number of 5-residue plus one secondary structure element segments), and $n = |S|$ (the number of attributes, $n = 5$ in this case).

As shown in Fig. (**1**), for a protein amino acid sequence and corresponding secondary structure sequence of length k, only the secondary structure elements from the third position to position $(k\text{-}2)$ are extracted as the 5-residue segments. In other words, the first and second positions at the beginning of the secondary structure sequence, as well as the last and second-to-last positions at the end of the secondary structure sequence, are not extracted as 5-residue segments. To handle these positions, extractions are done slightly differently, as shown in Fig. (**2**).

These 3-residue and 4-residue segments also are used as input to the RT-RICO rule generation algorithm to generate rules. As previously mentioned, the input to RT-RICO Step 2, Rule Generation, is in the form of an $m\times(n+1)$ matrix, where m is the number of all entities, and $n = |S|$ (the number of attributes, where $n = 3$ for 3-residue segments, and $n=4$ for 4-residue segments). The same rule generation algorithm applies to all these segments. The rules generated are used in step 3 to predict the secondary structure elements at the first and second positions, as well as the last and second-to-last positions of unknown secondary structure sequences, respectively.

For an amino acid sequence of length k, $(k\text{-}4)$ 5-residue segments are extracted, whereas only two 3-residue segments (in the first and last positions), and two 4-residue segments (in the second and second-to-last positions) are extracted. As the extraction is done for a large number of protein domains (Table 1), the rule generation and prediction operations in later steps involve mostly 5-residue segments in terms of the training data size. Due to this reason, only 5-residue segment numbers are recorded in the prediction result tables, and only 5-residue segment numbers are considered in the algorithm time complexity that is discussed in later sections.



Protein Name: 1rte:A
Primary Structure: MGLLSRLRKREPISIYDK...
Secondary Structure: --HHHHHHCC----HHHH...
Protein Name: 1s69:A
Primary Structure: MSTLYEKLGGTTAVDLAV...
Secondary Structure: ---HHHHHCHHHHHHHHH...
Protein Name: 1s6a:A
Primary Structure: MSTLYEKLGGTTAVDLAV...
Secondary Structure: ---HHHHHCHHHHHHHHH...
....
3 amino acid residues + 1 secondary structure element segments:
M,G,L,-
M,S,T,-
M,S,T,-
....

Protein Name: 1rte:A
Primary Structure: MGLLSRLRKREPISIYDK...
Secondary Structure: --HHHHHHCC----HHHH...
Protein Name: 1s69:A
Primary Structure: MSTLYEKLGGTTAVDLAV...
Secondary Structure: ---HHHHHCHHHHHHHHH...
Protein Name: 1s6a:A
Primary Structure: MSTLYEKLGGTTAVDLAV...
Secondary Structure: ---HHHHHCHHHHHHHHH...
....
4 amino acid residues + 1 secondary structure element segments:
M,G,L,L,-
M,S,T,L,-
M,S,T,L,-
....

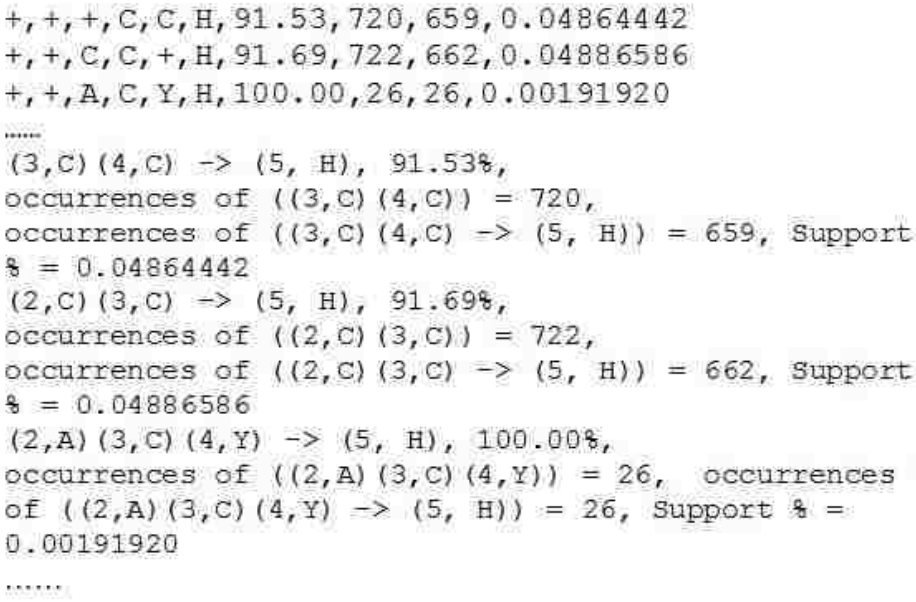
Note: The last and second-to-last positions at the end of the sequences are also represented by 3 residues + 1, and 4 residues + 1 segments, respectively. The segments are generated in a similar way, but form separate training datasets.

Fig. (2). Protein primary structure 3-residue segments and related secondary structure elements representation, protein primary structure 4-residue segments and related secondary structure elements representation, at the beginning of the sequences.

3.3. RT-RICO Step 2, Rule Generation

RT-RICO generates rules based on the segments in the form of an *m×(n+1)* matrix. The main RT-RICO rule generation algorithm is covered in Section 4. Some examples of the generated rules are shown in Fig. (**3**) in two separate formats. The first format is intended to be read by the computer programs at the later prediction stage (i.e., the computer rule format). The second format is intended to be read by the user (i.e., the human rule format). The first rule (in human rule format) is interpreted as follows: if the fourth position attribute (or “3” as interpreted by program) is “C”, and the fifth position attribute (or “4” as interpreted by program) is “C”, then the sixth attribute (decision attribute, or “5” as interpreted by program) is “H” with a confidence of 91.53% and a support of 0.04864442%. The definitions of confidence and support can be found in [27].

The corresponding first rule (in computer rule format) is interpreted as follows: if the first position attribute is “+” (representing any amino acid element), the second position attribute is “+”, the third position attribute is “+”, the fourth position attribute is “C”, and the fifth position attribute is “C”, then the sixth attribute (i.e., the decision attribute) is “H.” The number of occurrences of the fourth position attribute (which is “C”) and the fifth position attribute (which is “C”) equals 720 among all inputs to RT-RICO. The number of occurrences of the fourth position attribute (which is “C”), the fifth position attribute (which is “C”), and the sixth attribute (which is “H”), equals 659 among all inputs to RT-RICO. The confidence is 91.53% and the support is 0.04864442%.

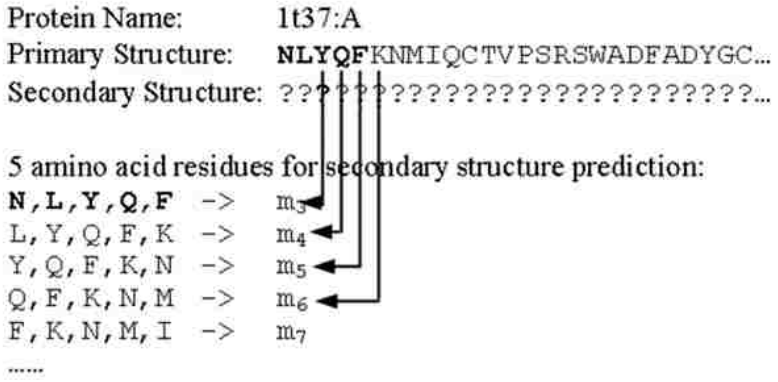


```
+,+,+,C,C,H,91.53,720,659,0.04864442
+,+,C,C,+,H,91.69,722,662,0.04886586
+,+,A,C,Y,H,100.00,26,26,0.00191920
......
(3,C)(4,C) -> (5, H), 91.53%,
occurrences of ((3,C)(4,C)) = 720,
occurrences of ((3,C)(4,C) -> (5, H)) = 659, Support
% = 0.04864442
(2,C)(3,C) -> (5, H), 91.69%,
occurrences of ((2,C)(3,C)) = 722,
occurrences of ((2,C)(3,C) -> (5, H)) = 662, Support
% = 0.04886586
(2,A)(3,C)(4,Y) -> (5, H), 100.00%,
occurrences of ((2,A)(3,C)(4,Y)) = 26,  occurrences
of ((2,A)(3,C)(4,Y) -> (5, H)) = 26, Support % =
0.00191920
......
```

Fig. (3). Sample rules generated by RT-RICO.

3.4. RT-RICO Step 3, Prediction

Finally RT-RICO loads protein primary structures from the test dataset, and predicts the secondary structure elements. As shown in Fig. (**4**), for each secondary structure element prediction position (for a corresponding amino acid sequence of length *k*, from position *3* to *k-2*), five “neighboring” amino acid residues are extracted to form a segment of five amino acid residues. Each of these segments is compared with the generated rules (generated from 5-residue segments). If a segment matches a rule, the support value of the rule is taken into consideration for the prediction of the related secondary structure element.



Protein Name: 1t37:A
Primary Structure: **NLYQF**KNMIQCTVPSRSWADFADYGC...
Secondary Structure: ???????????????????????????...

5 amino acid residues for secondary structure prediction:
N,L,Y,Q,F -> m_3
L,Y,Q,F,K -> m_4
Y,Q,F,K,N -> m_5
Q,F,K,N,M -> m_6
F,K,N,M,I -> m_7
......

Fig. (4). Protein primary structure 5-residue segments and related secondary structure elements prediction. m_i is an element of set {H,E,C,-}. It is then converted to an element of the set {H, E, C}. Note: The first and second positions at the beginning of the sequence are represented (i.e., predicted) by 3-residue, and 4-residue segments, respectively.

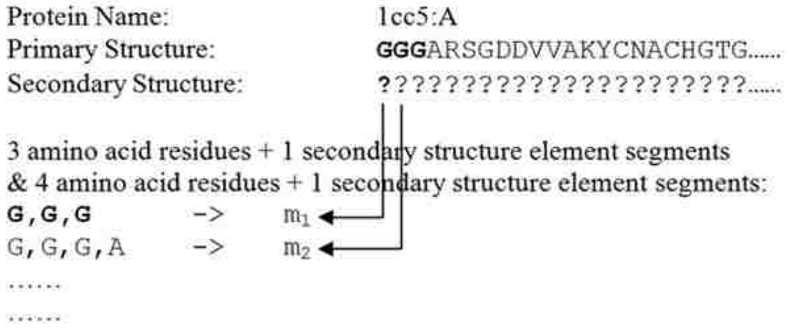
The algorithm first searches for matching rules with 100% confidence value. The secondary structure element with the highest total support value (among 100% confidence value rules) is selected.

If no matching rule exists among 100% confidence value rules, the algorithm then searches for other matching rules (with confidence values greater than or equal to 90%, but less than 100%). The secondary structure element with the highest total support value among these rules is selected as the predicted secondary structure element for that specific position.

If no matching rule is found for the segment at all, the secondary structure of the previous position is used as the predicted secondary structure.

To predict the first and second positions at the beginning of a secondary structure sequence, and the last and second-to-last positions at the end of a secondary structure sequence, three or four “neighboring” amino acid residues are extracted, as shown in Fig. (**5**). The same prediction algorithm mentioned above is responsible for the secondary structure prediction at these positions, but instead using rules generated from 3-residue and 4-residue segments as was discussed in Section 3.2.

The number of residues used in the RS126 test dataset, and the final Q_3 score of the RS126 set are shown in Table 1.



Protein Name: 1cc5:A
Primary Structure: **GGG**ARSGDDVVAKYCNACHGTG......
Secondary Structure: **?**?????????????????????......

3 amino acid residues + 1 secondary structure element segments
& 4 amino acid residues + 1 secondary structure element segments:
G,G,G -> m_1
G,G,G,A -> m_2
......
......

Fig. (5). Protein primary structure 3-residue, 4-residue segments, and related secondary structure elements prediction. m_i is an element of set {H,E,C,-}. It is then converted to an element of the set {H, E, C}. Note: The last and second-to-last positions at the end of the sequence are also represented (i.e., predicted) by 3-residue, and 4-residue segments, respectively.

4. Main RT-RICO Rule-Generation Algorithm

Although the RT-RICO protein secondary structure prediction method consists of the three steps mentioned in Section 3, the most computationally intensive part is in the second step, rule generation. This section covers the details of that algorithm.

4.1. Rule Induction From Coverings

RT-RICO is based on a previously implemented method called RICO (Rule Induction from Coverings) [20]. RICO uses some of the concepts introduced by Pawlak

[28] for rough sets, a classification scheme based on partitions of entities in a dataset [29].

In this approach, if S is a set of attributes and R is a set of decision attributes (i.e., attributes whose values we are interested in being able to determine if the values of the attributes in the set S are known), then a covering P of R in S can be found if the following three conditions are satisfied:

i. P is a subset of S.

ii. *R depends* on P (i.e., *P determines R*). That is, if a pair of entities x and y cannot be distinguished by means of attributes from P, then x and y also cannot be distinguished by means of attributes from R. If this is true, then entities x and y are said to be *indiscernible* by P (and, hence, R), denoted $x \sim_P y$. An *indiscernibility relation* $\sim_P$ is such a partition over all entities in the data set.

iii. P is minimal.

Condition (ii) is true if and only if an equivalent condition ≤, known as the *attribute dependency inequality*, holds for P^* and R^*, the partitions of all attributes and decisions generated by P and R, respectively, where, for a set of attributes A:

$$A^* = \prod_{a \in A} \sim [a]^*$$

The inequality $P^* \leq R^*$ holds if and only if for each block B of P^*, there exists a block B' of R^* such that B is a subset of B'.

Once a covering is found, it is a straightforward process to induce rules from it. For example, if a set of attributes $P = \{a_1, a_2\}$ is found to determine a set of attributes $R = \{a_3\}$ (i.e., P is a covering for R), then rules of the form $(a_1, v_1) \wedge (a_2, v_2) \rightarrow (a_3, v_3)$ (read as “if a_1 equals v_1 and a_2 equals v_2, then a_3 equals v_3”) can be generated where v_1, v_2, and v_3 are actual values of attributes a_1, a_2, and a_3, respectively, for which the relationship holds in the dataset. Such a rule also conveys a notion of non-independence between the attributes in the sets P and R (e.g., a_3 is not independent of a_1 and a_2). Here non-independence means that the relationship between the two attributes could be correlation, dependency, or co-dependency.

4.2. Relaxed Attribute Dependency Inequality

All rules generated from coverings in this manner are “perfect” in the sense that there is no instance in the dataset for which the rule is not true. In order to relax this restriction somewhat (in much the same way that rules generated by decision tree induction are not always true for all instances in the dataset), the definition of the attribute dependency inequality can be modified as follows.

Definition 1: Relaxed Attribute Dependency Inequality

The inequality $P^* \leq_r R^*$ holds if and only if *there exists* a block B of P^*, and *there exists* a block B' of R^* such that B is a subset of B'.

As an example for the data set of Table 2, let $P = \{2\}$ and $R = \{3\}$. Then

$\{2\}^* = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6\}\}$

$\{3\}^* = \{\{x_1, x_2\}, \{x_3, x_5, x_6\}, \{x_4\}\}$

There exists a block $B = \{x_1, x_2\}$ in $\{2\}^*$ and a block $B' = \{x_1, x_2\}$ in $\{3\}^*$ such that $B \subseteq B'$. Thus, $\{2\}^* \leq_r \{3\}^*$ which means that $\{3\}$ depends on $\{2\}$ (i.e., $\{2\} \rightarrow_r \{3\}$) for at least *some* values of $\{2\}$. More specific rules can then be deduced from this relationship, such as $(2, D) \rightarrow (3, H)$.

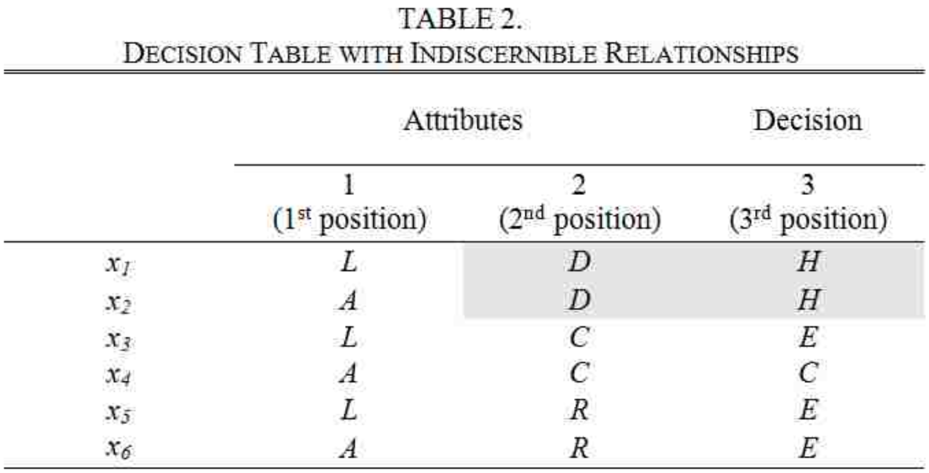


TABLE 2.
DECISION TABLE WITH INDISCERNIBLE RELATIONSHIPS

	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*L*	*D*	*H*
x_2	*A*	*D*	*H*
x_3	*L*	*C*	*E*
x_4	*A*	*C*	*C*
x_5	*L*	*R*	*E*
x_6	*A*	*R*	*E*

4.3. Relaxed Coverings

Similarly, the definition of a covering can be relaxed in order to induce rules depending on as small a number of attributes as possible.

Definition 2: Relaxed Coverings

A subset *P* of the set *S* is called a *relaxed covering* of *R* in *S* if and only if $P \rightarrow_r R$ and *P* is minimal in *S*. This is equivalent to saying that a subset *P* of the set *S* is a *relaxed covering* of *R* in *S* if and only if $P \rightarrow_r R$ and no proper subset *P'* of *P* exists such that $P' \rightarrow_r R$.

As an example for the dataset of Table 2, suppose rules need to be induced for *R = {3}*. The covering *{1, 2}* can be used; that is, for any assignment of values for the covering *{1, 2}*, each entity in Table 2 will induce a rule for *{3}*. But, instead of inducing a rule by looking at combinations of values for *{1, 2}*, such as $(1, L) \wedge (2, D) \rightarrow (3, H)$, rules are induced based on values for only *{1}* or *{2}*. Thus, $(2, D) \rightarrow (3, H)$ will be generated as a rule since $\{2\} \rightarrow_r \{3\}$ and *{2}* is minimal in *{1, 2}*. In this manner, *{2}* is a *relaxed covering* of *{3}*.

4.4. Checking Attribute Dependency

To implement rule induction from coverings with the relaxed constraints, it is necessary to use the concept of checking attribute dependency, which was introduced by Grzymala-Busse [29]. In order for *P* to be a relaxed covering of *R* in *S*, the following conditions must be true:

i. *P* must be a subset of *S*,
ii. *R* must depend on set *P* (for some values of *P*), and
iii. *P* must be minimal.

For the specific application of generating rules for protein secondary structure prediction, rules involving more attributes are preferred over rules involving fewer attributes, because they normally generate higher confidence values. In addition, all the possible attribute position combinations are needed to predict secondary structure. As a result, condition (iii) is not enforced for rule generation in our implementation. In fact, condition (iii) cannot be enforced for this particular application; otherwise, many meaningful rules involving multiple attributes and high confidence values would not be generated, leading to inaccurate predictions.

Condition (ii) is true if and only if the relaxed attribute dependency inequality, $P^* \leq_r R^*$, is satisfied.

The question then becomes how the above inequality can be efficiently checked. For each set P, a new partition, generated by P, must be determined. Partition U should be generated by P. For partitions π and τ of U, $\pi \cdot \tau$ is a partition of U such that two entities, x and y, are in the same block of $\pi \cdot \tau$ if and only if x and y are in the same block for both partitions π and τ of U. For example, referring to Table 3,

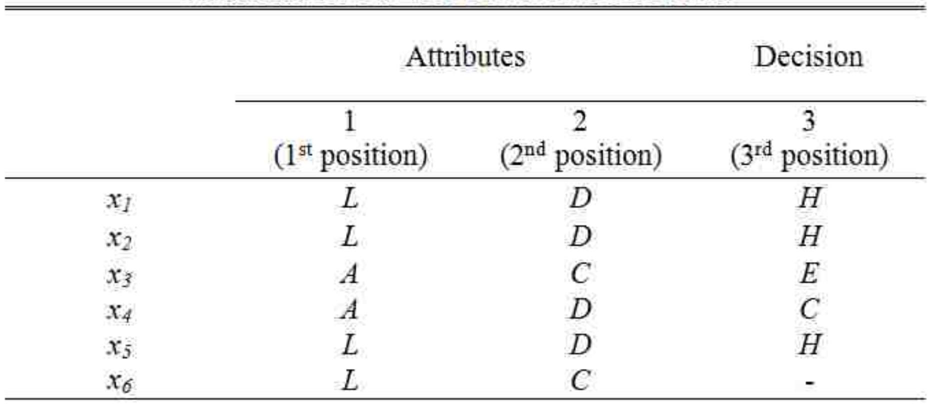
$\{1\}^* = \{\{x_1, x_2, x_5, x_6\}, \{x_3, x_4\}\}$

$\{2\}^* = \{\{x_1, x_2, x_4, x_5\}, \{x_3, x_6\}\}$

$\{1\}^* \cdot \{2\}^* = \{\{x_1, x_2, x_5\}, \{x_3\}, \{x_4\}, \{x_6\}\}$

That is, for $\{1\}^*$ and $\{2\}^*$, two entities x_1 and x_2 are in the same block of $\{1\}^* \cdot \{2\}^*$ if and only if x_1 and x_2 are in the same block of $\{1\}^*$ and in the same block of $\{2\}^*$. Further, the relaxed covering of $\{3\}$ is $\{1, 2\}$, because $\{1\}^* \cdot \{2\}^* \leq_r \{3\}^*$, and $\{1, 2\}$ is minimal since $\{1\}^* \leq_r \{3\}^*$ and $\{2\}^* \leq_r \{3\}^*$ are both not true.

TABLE 3.
DECISION TABLE WITH RELAXED COVERING



	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*L*	*D*	*H*
x_2	*L*	*D*	*H*
x_3	*A*	*C*	*E*
x_4	*A*	*D*	*C*
x_5	*L*	*D*	*H*
x_6	*L*	*C*	-

4.5. Finding the Set of All Relaxed Coverings

The algorithm R-RICO (Relaxed Rule Induction from Coverings) which is given below can be used to find the set C of all relaxed coverings of R in S (as well as the related rules).

Let S be the set of all attributes, and let R be the set of all decision attributes. Let k be a positive integer. The set of all subsets of the same cardinality k of the set S is denoted $P_k = \{\{x_{i1}, x_{i2}, \dots, x_{ik}\} \mid x_{i1}, x_{i2}, \dots, x_{ik} \in S\}$ [29].

Algorithm 1: R-RICO

```
begin
    for each attribute x in S do
        compute [x]*;
    compute partition R*
    k:=1
    while k ≤ |S| do
        for each set P in $P_k$ do
            if ($\prod_{x \in P}$ [x]* $\leq_r$ R*) then
            begin
                find the attribute values from the first block B of P
                and from the first block B' of R;
                add rule to output file;
            end
        k := k+1;
    end-while
end-algorithm.
```

Note that the condition (iii) for a relaxed covering is not enforced in the R-RICO algorithm. The time complexity of the R-RICO algorithm is exponential to |*S*|, the number of attributes in the dataset.

4.6. RT-RICO Algorithm

The R-RICO algorithm produces rules that are 100% correct. However, unlike decision tree induction, R-RICO produces a more comprehensive rule set. The algorithm can be further modified to satisfy some particular level of uncertainty in the rules (e.g., the rule is ≥ 50% true). That is, rather than just reporting a rule *R*, the rule can be reported as a tuple *(R, p)* where *p* is the probability that rule *R* is true. To accommodate this information in the rules, the definition of attribute dependency inequality must be further modified as in Definition 3.

Definition 3: Relaxed Attribute Dependency Inequality with Threshold

Set R depends on a set P with threshold probability t ($0 < t \leq 1$), and is denoted by $P \rightarrow_{r,t} R$ if and only if $P^* \leq_{r,t} R^*$ and there exists a block B of P^*, and there exists a block B' of R^* such that $(|B \cap B'| \,/\, |B|) \geq t$.

It can be observed that, when $t=1$, Definitions 1 and 3 represent the same mathematical relation.

As an example, for the dataset of Table 4, let $P = \{1, 2\}$, $R = \{3\}$, and $t = 0.6$. Then the following partitions can be formed:

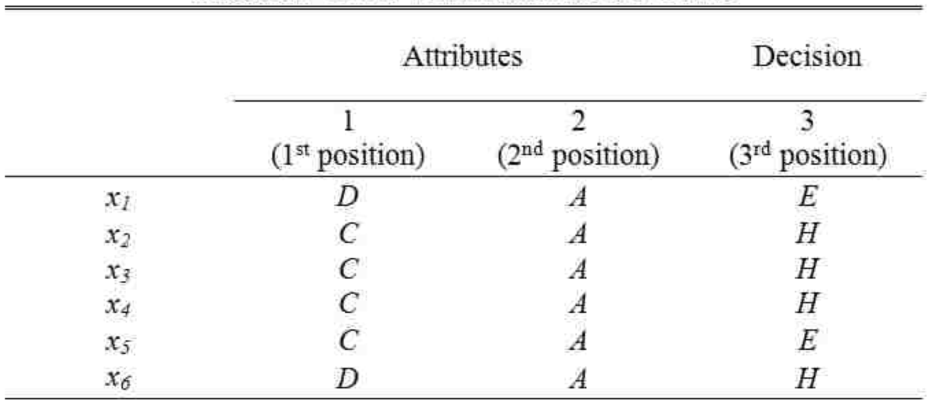
$\{1\}^* = \{\{x_1, x_6\}, \{x_2, x_3, x_4, x_5\}\}$

$\{2\}^* = \{\{x_1, x_2, x_3, x_4, x_5, x_6\}\}$

$P^* = \{1,2\}^* = \{1\}^* \cdot \{2\}^* = \{\{x_1, x_6\}, \{x_2, x_3, x_4, x_5\}\}$

$R^* = \{3\}^* = \{\{x_1, x_5\}, \{x_2, x_3, x_4, x_6\}\}$

TABLE 4.
DECISION TABLE WITH RELAXED COVERING



	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*D*	*A*	*E*
x_2	*C*	*A*	*H*
x_3	*C*	*A*	*H*
x_4	*C*	*A*	*H*
x_5	*C*	*A*	*E*
x_6	*D*	*A*	*H*

There exists a block $B = \{x_2, x_3, x_4, x_5\}$ in $\{1, 2\}^*$, and there exists a block $B' = \{x_2, x_3, x_4, x_6\}$ in $\{3\}^*$ such that $(|B \cap B'| \,/\, |B|) = |\{x_2, x_3, x_4\}| \,/\, |\{x_2, x_3, x_4, x_6\}| = 0.75 \geq 0.6$. Thus, $P^* = \{1, 2\}^* \leq_{r,t} R^* = \{3\}^*$, and $\{3\}$ depends on $\{1, 2\}$ (i.e., $\{1, 2\} \rightarrow_{r,t} \{3\}$), with threshold probability 0.6.

The corresponding values of attributes can be found from entities that are in $B \cap B' = \{x_2, x_3, x_4\}$ for the sets $P = \{1, 2\}$ and $R = \{3\}$; namely, the value of attribute 1 is C, the value of attribute 2 is A at $\{x_2, x_3, x_4\}$, and the value of decision 3 is H for entities $\{x_2, x_3, x_4\}$. The rule induced from $\{1, 2\} \rightarrow_{r,t} \{3\}$ is then $(1, C) \wedge (2, A) \rightarrow (3, H)$ with a probability (confidence) of 75%. Another way to look at this is to note that the number of occurrences of $((1,C)(2,A)) = 4$, and the number of occurrences of $((1,C)(2,A) \rightarrow (3, H)) = 3$.

The definition of relaxed coverings must also be modified to incorporate the notion of the threshold probability given in Definition 4.

Definition 4: Relaxed Coverings with Threshold Probability

Let *S* be a nonempty subset of a set of all attributes, and let *R* be a nonempty subset of decision attributes, where *S* and *R* are disjoint. A subset *P* of the set *S* is called a relaxed covering of *R* in *S* with threshold probability t $(0 < t \leq 1)$ if and only if $P \rightarrow_{r,t} R$ and *P* is minimal in *S*.

Algorithm RT-RICO (Relaxed Threshold Rule Induction From Coverings) finds the set *C* of all relaxed coverings of *R* in *S* (and the related rules), with threshold probability t $(0 < t \leq 1)$, where *S* is the set of all attributes, and *R* is the set of all decisions. The set of all subsets of the same cardinality *k* of the set *S* is denoted $P_k = \{\{x_{i1}, x_{i2}, \ldots , x_{ik}\} \mid x_{i1}, x_{i2}, \ldots , x_{ik} \in S\}$.

Algorithm 2: RT-RICO

```
begin
    for each attribute x in S do
        compute [x]*;
    compute partition R*
    k:=1
    while k ≤ |S| do
        for each set P in $P_k$ do
        if ($\prod_{x \in P}$ [x]* $\leq_{r,t}$ R*) then
        begin
            find values of attributes from the entities that are in the
            region (B ∩ B’) such that (|B ∩ B’| / |B|) ≥  t;
            add rule to output file;
        end
        k := k+1
    end-while;
end-algorithm.
```

Note that the condition “*P* is minimal in *S*” of a relaxed covering with threshold probability is not enforced in the RT-RICO algorithm. The reason for not implementing this condition is the same as the reason mentioned for the R-RICO algorithm. For this application, to generate rules for protein secondary structure prediction, rules involving more attributes are preferred over rules involving fewer attributes, because they normally generate higher confidence values. Also, all the possible attribute position combinations are needed for accurate prediction.

The time complexity of the RT-RICO algorithm is again exponential to |*S*|, the number of attributes in the dataset. Specifically, the time complexity is $O(m^2 2^n)$, where *m* is the number of all entities (the number of 5-residue segments), and $n = |S|$ (the number of attributes). It would appear that 2^n dominates the time complexity. But, for the training datasets of this application, $n = |S| = 5$, and *m* is sufficiently large. Hence, m^2 dominates the time complexity in this case.

As mentioned in Section 3, the rules generated by the RT-RICO algorithm are then compared with the proteins in the test dataset to predict the secondary structure elements.

5. Parallelized/Modified RT-RICO Algorithm

The RT-RICO algorithm has a time complexity of $O(m^2 2^n)$, where *m* is the number of all entities (the number of 5-residue segments), and $n = |S|$ (the number of attributes). In practice, *n* is only *5*, while *m* can be fairly large. Hence, m^2 dominates the time complexity. The test programs were written in PERL, and the largest *m* value tested was 137,715. When executed on a computer with an Intel Pentium Dual-Core processor, 2 GB of RAM, and Windows XP OS, the total program running time was approximately 14 days.

In order to accommodate a larger dataset (e.g., *m* value 3,366,832), two new algorithms (Modified RT-RICO and Parallelization of Modified RT-RICO) were developed. The time complexity of modified RT-RICO is only $O(m2^n)$, although it comes at an acceptable sacrifice of space complexity (i.e., more main memory space is needed, as is discussed later in this section). The program was parallelized using an NVIDIA Tesla C1060 GPU with 4GB of RAM. The 240 cores on this GPU each run at 1.3 GHz.

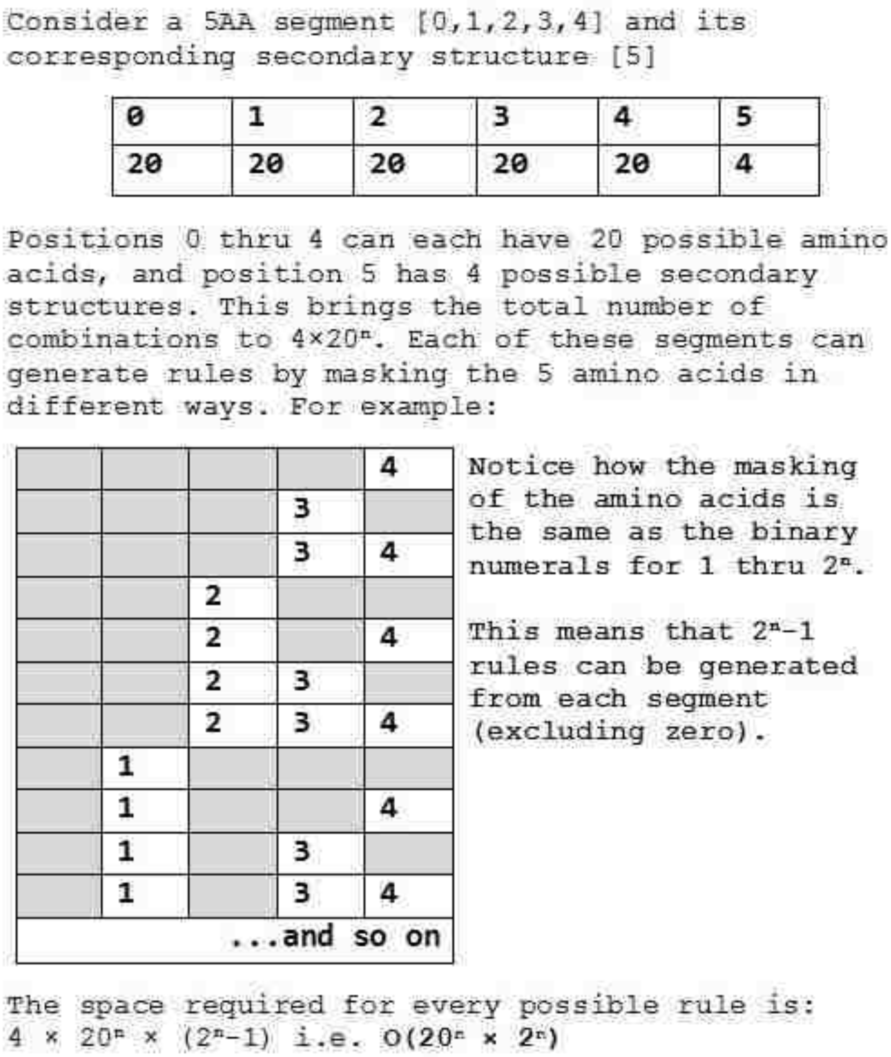
The CPU on the same test machine is a 4-core Intel Core i7-920 with 8GB of RAM. With the modified algorithm, and the new hardware, the total program running time improved from days to a few minutes.

The focus of the parallelization of RT-RICO was the rule generation step. It is the most expensive part of the algorithm since it involves generating rules from each segment, counting the frequency of each rule, and finally calculating the confidence and support of each rule. As mentioned earlier, in the sequential implementation of RT-RICO, the complexity of this step is $O(m^2 2^n)$, where m is the number of segments and n is the number of amino acid residues in a segment. Usually n is fixed at 5, but m could range from a few thousand to the millions. To reduce the complexity, and hence improve its running time, it was essential to reduce the factor of m in the RT-RICO algorithm.

The m^2 in $O(m^2 2^n)$ is a result of counting the occurrences of each rule. After generating a rule from a segment, the algorithm has to iterate through the list of m segments to count how many times that rule has been seen. This has to be repeated for each of the $m2^n$ rules that can be generated. Hence the complexity is $O(m^2 2^n)$.

But RT-RICO can skip the iteration through the list m times per rule if it simply increments a rule-specific counter every time a rule is generated. The drawback is that there needs to be a counter for every possible rule that can be generated, and this requires an immense amount of main memory. In the worst-case, $20^n \times 2^n$ rules can be generated, which translates to approximately 99 Megabytes for 5aa segments, and 163 Gigabytes for 7aa segments. This increases exponentially with an increase in n. The calculation of space complexity is illustrated in Fig. (**6**).

Despite the exponential space complexity, 5aa segments only require 99 Megabytes of memory. This was further reduced to just 4 Megabytes, by accounting for the duplicate rules that two different segments can generate. For example, the two 5aa segments [S,L,F,E,Q] and [E,L,S,E,Q] can generate the same rule for [+,L,+,E,Q]. The mathematics behind this space optimization is not explained here, because the 99 MB, or the 4 MB required by the modified algorithm, are both trivial amounts on the newer test machine that was used (which has 8192 Megabytes of memory).



Consider a 5AA segment [0,1,2,3,4] and its corresponding secondary structure [5]

0	1	2	3	4	5
20	20	20	20	20	4

Positions 0 thru 4 can each have 20 possible amino acids, and position 5 has 4 possible secondary structures. This brings the total number of combinations to 4×20^n. Each of these segments can generate rules by masking the 5 amino acids in different ways. For example:

				4
			3	
			3	4
		2		
		2		4
		2	3	
		2	3	4
	1			
	1			4
	1		3	
	1		3	4
...and so on				

Notice how the masking of the amino acids is the same as the binary numerals for 1 thru 2^n.

This means that 2^n-1 rules can be generated from each segment (excluding zero).

The space required for every possible rule is: $4 \times 20^n \times (2^n-1)$ i.e. $O(20^n \times 2^n)$

Fig. (6). The number of all possible rules from 5aa segments.

5.1. Modified Algorithm for Rule Generation

In essence, the modified RT-RICO algorithm compromises on space complexity for the sake of reducing time complexity. Algorithm 3 describes this modification in more detail.

Algorithm 3: Modified RT-RICO

```
begin
    Allocate counters for every possible rule (initialize to 0)
    for each segment
```

```
            for each $2^n$-1 rules that can be generated from this segment
                Calculate the memory location of the counter
                corresponding to this rule, and increment it by 1
            end-for
        end-for
        Read each counter and calculate the confidence and support for those rules
        that pass the relaxed threshold
    end-algorithm.
```

The complexity of this algorithm is just $O(m2^n)$ because the algorithm does not need to count the reoccurrence of each rule. The generated rules simply increment a counter whenever they are generated. There is an additional amount of time required to calculate the memory location of the counter that corresponds to a rule. However, this is negligible, and as a constant, it does not affect the overall complexity of the algorithm.

5.2. Parallelization of Rule Generation

The modified RT-RICO rule generation algorithm places no restrictions on the order in which rules are generated. So parallelizing the algorithm involves a straightforward distribution of the input data among processing units. Each processing unit accepts a segment as input, determines a rule from that segment, and increments the shared memory counter corresponding to that rule. Theoretically, these operations can be performed in parallel by any number of concurrent processing units. However, to minimize potentially conflicting concurrent updates of shared memory locations, the number of concurrent processing units (p) is kept at 2^n-1, which is the number of rules that a single segment can generate. Since these 2^n-1 rules are guaranteed to be distinct, they would guarantee mutually exclusive concurrent updates of shared memory counters. Algorithm 4 shows a parallelized version of Algorithm 3. The time complexity of Algorithm 4 is $O((m2^n)/p)$, where p equals the number of concurrent processing units.

```
    **Algorithm 4**: Modified  RT-RICO
    begin
```

```
        Allocate counters for every possible rule (initialize to 0)
        for each segment s
                Send s to $2^n$-1 processes that each calculates a different rule from
                it, and increment the corresponding shared memory counter
        end-for
        Read each counter and calculate the confidence and support for those rules
        that pass the relaxed threshold
end-algorithm.
```

5.3. Massively Parallel Computation Using GPUs

Compute Unified Device Architecture (CUDA) is a programming interface for developing general purpose applications on Graphics Processing Units (GPUs). GPUs are conventionally used for graphics acceleration, which typically involves repeatedly performing the same computational operation on multiple input data, also known as SIMD (single instruction multiple data) operations. Because of the constraints placed on SIMD operations, GPU hardware is designed with features such as massively parallel processing and pipelining to accelerate the execution of these operations. With CUDA, GPUs can be directly programmed using the C programming language to process any kind of general purpose operation, which normally would be tasked to CPUs. However, because the GPU hardware remains the same, they are still ideally suited for SIMD operations, and more complex operations are likely to run faster sequentially on a CPU.

The modified RT-RICO rule generation algorithm is an ideal SIMD operation. The calculation of the memory location of the counter that corresponds to a rule extracted from a segment is performed over and over again for all the given segments in the input file. This SIMD operation was parallelized using an NVIDIA Tesla C1060 GPU with 4GB of RAM. The 240 cores on this GPU each run at 1.3 GHz. The CPU on the same test machine was a 4-core Intel Core i7-920 with 8GB of RAM. The total program running time was close to 3 minutes for rule generation of the dataset in Table 1.

6. Results

The RS126 set [4] and the CB396 set [2] are both non-redundant test datasets created with the objective of comparing different protein secondary structure prediction methods.

These two standard test datasets were used to evaluate the performance of the RT-RICO protein secondary prediction method. The two datasets have been studied extensively in other literature, and have been used as standard datasets to evaluate other prediction methods. Some of the prediction scores with different methods for the same datasets are mentioned in Sections 1 and 2. It should be noted that the CB396 set does not include protein domains from the RS126 set.

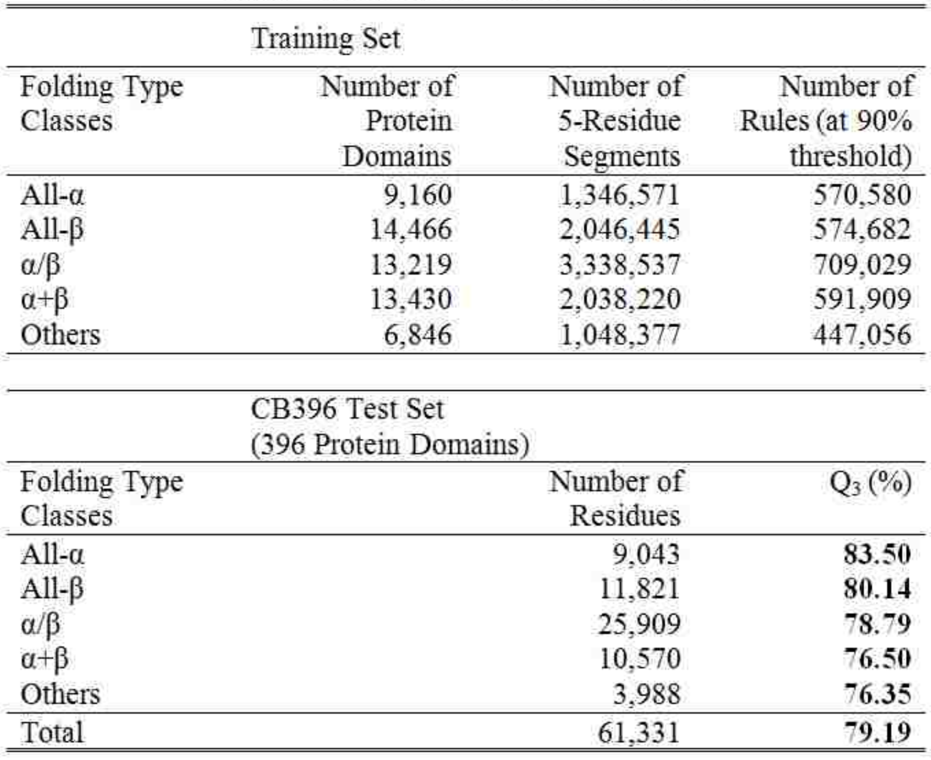
Table 1 lists the number of protein domains in each training dataset and the performance of the RT-RICO prediction method on the RS126 test dataset. Table 5 shows the number of protein domains in each training dataset and the performance of the RT-RICO on the CB396 test dataset.

Cuff and Barton [2] tested the RS126 set with various prediction methods and generated Q_3 scores of 73.5% (PHD), 71.1% (DSC), 70.3% (PREDATOR), 72.7% (NNSSP) and 74.8% for the CONSENSUS method. The final Q_3 scores of RT-RICO prediction using the RS126 test dataset are shown in Table 1. The “all-α” protein domains have the highest Q_3 score of 87.40%. The “all-β” and “α/β” protein domains have Q_3 scores of 82.22% and 78.05%, respectively. The “α+β” and “Others” protein domains have the prediction accuracy of 84.64% and 81.23%. On average, RT-RICO has a Q_3 score of 81.75%, which is higher than the Q_3 score generated by other methods using the same RS126 test dataset reported in [2].

Cuff and Barton [2] also tested the same prediction methods using the CB396 set, resulting in Q_3 scores of 71.9% (PHD), 68.4% (DSC), 68.6% (PREDATOR), 71.4% (NNSSP) and 72.9% for the CONSENSUS method. The final Q_3 scores of the RT-RICO prediction method on the CB396 test dataset are shown in Table 5. The “all-α” protein domains have the highest Q_3 score of 83.50%. The “all-β” and “α/β” protein domains have Q_3 scores of 80.14% and 78.79%, respectively. The “α+β” and “Others” protein domains have the prediction accuracy of 76.50% and 76.35%. On average, RT-RICO has a Q_3 score of 79.19%, which is higher than the Q_3 score generated by other methods using the same CB396 test dataset reported in [2].

Due to the different approaches and test designs, it should be noted that it is difficult to directly compare prediction results between this method and other methods. The final Q_3 scores comparison should be used as a general guide, not a strict percentile comparison.

TABLE 5.
PROTEIN SECONDARY STRUCTURE PREDICTION USING RT-RICO RULE GENERATION ON CB396 TEST DATASET



	Training Set		
Folding Type Classes	Number of Protein Domains	Number of 5-Residue Segments	Number of Rules (at 90% threshold)
All-α	9,160	1,346,571	570,580
All-β	14,466	2,046,445	574,682
α/β	13,219	3,338,537	709,029
α+β	13,430	2,038,220	591,909
Others	6,846	1,048,377	447,056

	CB396 Test Set (396 Protein Domains)	
Folding Type Classes	Number of Residues	Q_3 (%)
All-α	9,043	**83.50**
All-β	11,821	**80.14**
α/β	25,909	**78.79**
α+β	10,570	**76.50**
Others	3,988	**76.35**
Total	61,331	**79.19**

7. Conclusions

A novel rule-based method, RT-RICO, which generates rules that can be used in predicting protein secondary structure was presented in this paper. This method performed very well with the standard test datasets RS126 and CB396. The Q_3 scores of 81.75% for the RS126 set and 79.19% for the CB396 set are better than the Q_3 scores generated by comparable computational methods using the same datasets.

The main RT-RICO rule generation algorithm has a time complexity of $O(m^2 2^n)$, where m is the number of segments, with m^2 dominating the time complexity. The time complexity of the modified RT-RICO algorithm is only $O(m2^n)$ with m dominating the time complexity, although it comes at an acceptable sacrifice of space complexity. The time complexity of the parallelized RT-RICO algorithm is $O((m2^n)/p)$ where p is equal to the number of concurrent processing units.

The resulting fast running time of the program enables us to generate rules from the large amount of available protein data within an acceptable timeframe, and to predict the secondary structure of available test datasets efficiently. In the future, we plan to continue to look for ways to improve the accuracy of this new promising rule-based prediction method.

References

[1] B. Rost, “Rising accuracy of protein secondary structure prediction,” in *Protein structure determination, analysis, and modeling for drug discovery*, (ed. D Chasman), New York: Dekker, 2003, pp. 207–249.

[2] J. A. Cuff, and G. Barton, “Evaluation and improvement of multiple sequence methods for protein secondary structure prediction,” *Proteins*, vol. 34, pp. 508–519, 1999.

[3] W. Kabsh, and C. Sander, “How good are predictions of protein secondary structure?”, *FEBS Letters*, vol. 155, pp. 179-182, 1983.

[4] B. Rost, and C. Sander, “Prediction of protein secondary structure at better than 70% accuracy,” *J Mol Biol.*, vol. 232, pp. 584-599, 1993.

[5] R. D. King, and M. J. E. Sternberg, “Identification and application of the concepts important for accurate and reliable protein secondary structure prediction,” *Protein Sci.*, vol. 5, pp. 2298–2310, 1996.

[6] D. Frishman, and P. Argos, “Seventy-five percent accuracy in protein secondary structure prediction,” *Proteins*, vol. 27, pp. 329–335, 1997.

[7] A. A. Salamov, and V. V. Solovyev, “Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments,” *J Mol Biol.*, vol. 247, pp. 11–15, 1995.

[8] U. Y. Fadime, Y. O¨zlem, and T. Metin, “Prediction of secondary structures of proteinsnext term using a two-stage method,” *Computers & Chemical Engineering*, vol. 32(1-2), pp. 78-88, 2008.

[9] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The Protein Data Bank,” *Nucleic Acids Res.*, vol. 28(1), pp. 235-242, 2000.

[10] D. T. Jones, “Protein secondary structure prediction based on position-specific scoring matrices,” *J Mol Biol.*, vol. 292(2), pp. 195-202, 1999.

[11] K. Bryson, L. J. McGuffin, R. L. Marsden, J. J. Ward, J. S. Sodhi, and D. T. Jones, “Protein structure prediction servers at University College London,” *Nucleic Acids Res.*, vol. 33(Web Server issue), pp. W36-38, 2005.

[12] 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.*, vol. 25(17), pp. 3389-3402, 1997.

[13] J. A. Cuff, and G. Barton, “Application of multiple sequence alignment profiles to improve protein secondary structure prediction,” *Proteins*, vol. 40(3), pp. 502-511, 2000.

[14] B. Rost, and C. Sander, “Improved prediction of protein secondary structure by use of sequence profiles and neural networks,” *Proc. Natl. Acad. Sci. USA*, vol. 90, pp.7558–7562, 1993.

[15] H. Hu, Y. Pan, R. Harrison, and P. Tai, “Improved protein secondary structure prediction using support vector machine and a new encoding scheme and an advanced tertiary classifier,” *IEEE Trans. NanoBiosci.*, vol. 3, pp. 265–271, 2004.

[16] H. Kim, and H. Park, “Protein secondary structure prediction based on an improved support vector machines approach,” *Protein Eng.*, vol. 16, pp. 553-560, 2003.

[17] N. Nguyen, and J. C. Rajapakse, “Two stage support vector machines for protein secondary structure prediction,” *Intl J Data Mining & Bioinformatics,* vol. 1, pp. 248-269, 2007.

[18] M. Levitt, and C. Chothia, “Structural patterns in globular proteins,” *Nature*, vol. 261(5561), pp. 552-558, 1976.

[19] A. M. Maglia, J. L. Leopold, and V. R. Ghatti, “Identifying Character Non-Independence in Phylogenetic Data Using Data Mining Techniques,” in Proc Second Asia-Pacific Bioinformatics Conference Dunedin, New Zealand, 2004.

[20] J. L. Leopold, A. M. Maglia, M. Thakur, B. Patel, and F. Ercal, “Identifying Character Non-Independence in Phylogenetic Data Using Parallelized Rule Induction From Coverings,” in *Data Mining VIII: Data, Text, and Web Mining and Their Business Applications, WIT Transactions on Information and Communication Technologies,* vol. 38, pp. 45-54, 2007.

[21] W. Kabsch, and C. Sander, “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features,” *Biopolymers*, vol. 22 (12), pp. 2577–2637, 1983.

[22] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: an overview,” *Bioinformatics,* vol. 16(5), pp. 412-424, 2000.

[23] C. T. Zhang, and R. Zhang, “Q9, a content-balancing accuracy index to evaluate algorithms of protein secondary structure prediction,” *Int J Biochem Cell Biol.,* vol. 35(8), pp. 1256-1262, 2003.

[24] A. Andreeva, D. Howorth, J. M. Chandonia, S. E. Brenner, T. J. Hubbard, C. Chothia, and A. G. Murzin, “Data growth and its impact on the SCOP database: new developments,” *Nucleic Acids Res.*, vol. 36(Database issue), pp. D419-425, 2008.

[25] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, “SCOP: a structural classification of proteins database for the investigation of sequences and structures,” *J Mol Biol.*, vol. 247(4), pp. 536-540, 1995.

[26] J. L. Klepeis, and C. A. Floudas, “Ab initio prediction of helical segments in polypeptides,” *J Comput Chem.*, vol. 23(2), pp. 245-266, 2002.

[27] J. Han, and M. Kamber*, Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2001, pp. 155-157.

[28] Z. Pawlak, “Rough Classification,” *Int J Man-Machine Studies,* vol. 20, pp. 469-483, 1984.

[29] J. W. Grzymala-Busse, *Managing Uncertanity in Expert System,* Boston: Kluwer Academic, 1991, Ch 3.

PAPER

4. PROTEIN SECONDARY STRUCTURE PREDICTION USING BLAST AND RELAXED THRESHOLD RULE INDUCTION FROM COVERINGS

Leong Lee[1§], Jennifer L. Leopold [1], Ronald L. Frank[2]

[1] Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA

[2] Department of Biological Sciences, Missouri University of Science and Technology, Rolla, MO 65409 USA

[§]Corresponding author

Abstract

Background

Protein structure prediction has been a very important research problem in bioinformatics and biochemistry. The determination of protein structures by time-consuming and relatively expensive experimental methods is lagging far behind the explosive discovery of protein sequences. Despite the recent breakthrough of combining multiple sequence alignment information and artificial intelligence algorithms to predict protein secondary structure, the Q_3 accuracy of various computational prediction methods rarely has exceeded 75%; this status has changed little since 2003 when Rost stated that “the currently best methods reach a level around 77% three-state per-residue accuracy.”

Results

In this paper, a rule-based data-mining approach utilizing multiple sequence alignment information called BLAST-RT-RICO (Relaxed Threshold Rule Induction from Coverings) is presented. This method uses the PSI-BLAST algorithm to identify suitable proteins and generates association rules that can be used to predict protein secondary structure. This combined approach achieved a Q_3 score of 89.93% on the standard test dataset RS126 and a Q_3 score of 87.71% on the standard test dataset CB396, an improvement over comparable computational methods.

Conclusions

The current implementation of the BLAST-RT-RICO algorithm generates rules from the available protein data within an acceptable timeframe, efficiently predicting the protein secondary structure of test datasets. In the future, we plan to continue to look for ways to improve the accuracy of this new promising rule-based prediction method.

Background

Introduction

Prediction of the 3D structure of a protein from its amino acid sequence is a very important research goal in biochemistry and bioinformatics, and has been studied extensively since the 1960s. Protein structure prediction is valuable for drug design, enzyme design, and many other biotechnology applications. Rost [1] suggests that protein 3D structure prediction from sequence cannot be achieved fully; however, research has continuously improved methods for predicting simplified aspects of structure. Particularly in the area of secondary structure prediction, accuracy has surpassed the 70% threshold for all residues of a protein. That breakthrough was achieved by combining multiple sequence alignment information and artificial intelligence algorithms. Rost [1] also has stated that a value of around 88% likely will be the operational upper limit for prediction accuracy.

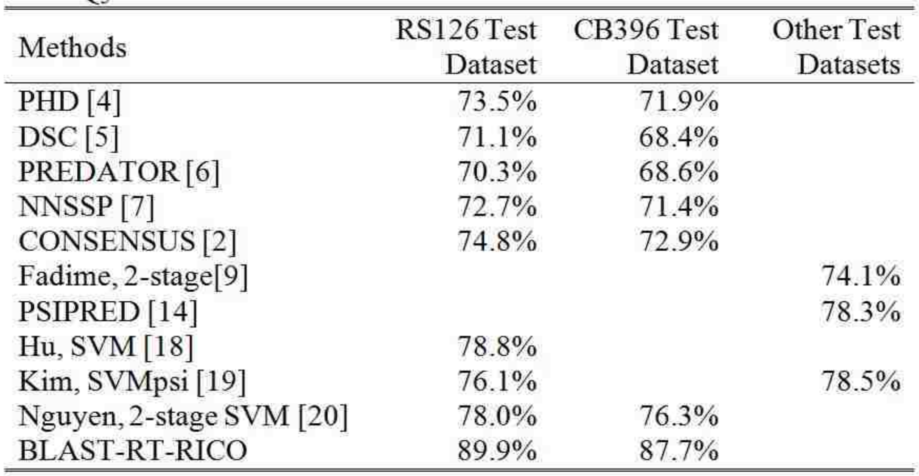
It is not an easy task to evaluate the performance of a protein secondary structure prediction method. For example, the use of different datasets for training and testing each algorithm makes it difficult to find an objective comparison of methods [2]. Interestingly, when Kabsh and Sanders [3] tested prediction methods using proteins that had not been used in the development of the algorithms, they found that the reported prediction accuracy of most of those methods decreased by more than 7%. One method's prediction accuracy decreased by as much as 27%. Rost [1] stated that "there is no value in comparing methods evaluated on different datasets."

Efforts have been made to develop standard test datasets to accurately evaluate the performance of prediction methods. Rost and Sander [4] selected a list of 126 protein domains (the RS126 set) that now constitutes a comparative standard. Cuff and Barton [2] described the development of a non-redundant test set of 396 protein domains (the CB396 set) where no two proteins in the set share more than 25% sequence identity over a length of more than 80 residues [4]. They used the CB396 set to test four secondary structure prediction methods: PHD [4], DSC [5], PREDATOR [6] and NNSSP [7]. They also combined the four methods by a simple majority-wins method, the CONSENSUS method [2]. The resulting Q_3 scores for the CB396 set were 71.9% (PHD), 68.4% (DSC), 68.6% (PREDATOR), 71.4% (NNSSP) and 72.9% for the CONSENSUS method. In the same research study, Cuff and Barton [2] also tested the RS126 set in which the Q_3 scores were 73.5% (PHD), 71.1% (DSC), 70.3% (PREDATOR), 72.7% (NNSSP) and 74.8% for the CONSENSUS method; see Table 1 for an overview of Q_3 scores of secondary structure prediction methods.

PHD, one of the first methods surpassing the 70% accuracy threshold, uses multiple sequence alignments as input to a neural network [8]. This approach effectively utilizes evolutionary information by exploiting the well-known fact that homologous proteins have similar 3D structures. Another interesting secondary structure prediction method described by Fadime, O¨zlem and Metin [9] uses a two-stage approach. In the first stage, the folding type of a protein is determined. The second stage utilizes data from the Protein Data Bank (PDB) [10] and a probabilistic search algorithm to determine the locations of secondary structure elements. The resulting average accuracy of their prediction score is 74.1%. This two-stage method shows that there are statistical relationships between a secondary structure element and its "neighboring" amino acid residues.

In this paper, we present a new method for predicting the secondary structure elements called BLAST-RT-RICO (Relaxed Threshold Rule Induction from Coverings). First, a query using the Web-based NCBI/PSI-BLAST search engine is performed for a protein [11]. Suitable proteins with significant multiple sequence alignments are identified. Then the algorithm, RT-RICO, generates rules for discovering dependencies between protein amino acid sequences and related secondary structure elements. These rules are used to predict protein secondary structure. The BLAST-RT-RICO method performed better than previously reported methods, with a Q_3 accuracy of 89.93% on the RS126 set and 87.71% on the CB396 set.

TABLE 1
Q_3 SCORES OF SECONDARY STRUCTURE PREDICTION METHODS



Methods	RS126 Test Dataset	CB396 Test Dataset	Other Test Datasets
PHD [4]	73.5%	71.9%	
DSC [5]	71.1%	68.4%	
PREDATOR [6]	70.3%	68.6%	
NNSSP [7]	72.7%	71.4%	
CONSENSUS [2]	74.8%	72.9%	
Fadime, 2-stage[9]			74.1%
PSIPRED [14]			78.3%
Hu, SVM [18]	78.8%		
Kim, SVMpsi [19]	76.1%		78.5%
Nguyen, 2-stage SVM [20]	78.0%	76.3%	
BLAST-RT-RICO	89.9%	87.7%	

Note: Due to the different approaches, different protein secondary structure data availability and different test design strategies, it is difficult to directly compare different methods' prediction results. The Q_3 scores comparison should be used as a general guide, not a strict percentile comparison.

Q_3 scores of PHD [4], DSC [5], PREDATOR [6], NNSSP [7] and CONSENSUS [2] are from the research paper of Cuff and Barton [2].

Q_3 scores under "Other Test Datasets" column should NOT be directly compared, because they use different test datasets.

Problem Description

In general, the protein secondary structure prediction problem can be characterized in terms of the following components [12]:

- Input

Amino acid sequence, $A = a_1, a_2, \dots a_N$

Data for comparison, $D = d_1, d_2, \dots d_N$

a_i is an element of a set of 20 amino acids, $\{A,R,N\dots V\}$

d_i is an element of a set of secondary structures, $\{H,E,C\}$, which represents helix H, sheet E, and coil C.

- Output

  Prediction result: $X = x_1, x_2, \ldots x_N$

  x_i is an element of a set of secondary structures, $\{H,E,C\}$

- 3-Class Prediction [13]

  This is a characterization of the problem as a multi-class prediction problem with 3 classes $\{H,E,C\}$ in which one obtains a 3 × 3 confusion matrix $Z = (z_{ij})$. z_{ij} represents the number of times the input is predicted to be in class j while belonging to class i.

  $Q_{total} = 100 \sum_i Z_{ii} / N$

- Q_3 Score

  Accuracy is computed as $Q_3 = W_{\alpha\alpha} + W_{\beta\beta} + W_{cc}$

  $W_{\alpha\alpha}$ = % of helices correctly predicted ($100\ Z_{11} / N$ or $100\ Z_{HH} / N$)

  $W_{\beta\beta}$ = % of sheets correctly predicted ($100\ Z_{22} / N$ or $100\ Z_{EE} / N$)

  W_{cc} = % of coils correctly predicted ($100\ Z_{33} / N$ or $100\ Z_{CC} / N$)

  In other words, a protein secondary structure data sequence D is compared to the predicted result sequence X to calculate the Q_3 score.

Related Work

Rost [1] classifies protein secondary structure prediction methods into three generations. The first generation methods depend on single residue statistics to perform prediction. The second generation methods depend on segment statistics. The third generation methods use evolutionary information to predict secondary structure. For example, PHD [4] is a third generation prediction method based on a multiple-level neural network approach.

One of the best secondary structure predictors is the PSIPRED Protein Structure Prediction Server [14], which was developed at University College London [14, 15]. PSIPRED uses a two-stage neural network to predict the protein's secondary structure based on position-specific scoring matrices generated by PSI-BLAST (Position-Specific Iterated BLAST) [16]. The PSIPRED's Q_3 score based on a set of 187 unique folds is between 76.5% and 78.3% [14]. There are other secondary structure prediction methods that utilize neural network prediction algorithms; for example, Jnet examines multiple sequence alignments alongside profiles such as PSI-BLAST and HMM [17].

An important consideration in many of these approaches is the knowledge that random mutations in DNA sequence can lead to different amino acids in the protein sequences. These changes are considered the basis of evolution; mutations resulting in a structural change are not likely to retain protein function. Thus, structure is more conserved than sequence [1]. All naturally evolved protein pairs that have 35 of 100 pairwise identical residues have similar structures [1]. This is the basis of how evolutionary information is used in the form of multiple sequence alignments for predicting protein secondary structure. For most neural network methods mentioned above, the inputs to the neural networks are not single sequences, but rather different forms of updated profiles generated from multiple sequence alignments.

Recently, there has been a trend to use the support vector machine (SVM) to predict protein secondary structures. Hu, Pan, Harrison and Tai [18] achieved a Q_3 accuracy of 78.8% on the RS126 dataset using a SVM approach. Kim and Park [19] developed the SVMpsi method that resulted in Q_3 scores of 76.1% on the RS126 dataset and 78.5% on their KP480 dataset. Nguyen and Rajapakse [20] proposed a two-stage multi-class SVM approach utilizing position-specific scoring matrices generated by PSI-BLAST; the resulting Q_3 scores were 78.0% on the RS126 dataset and 76.3% on the CB396 dataset.

Levitt and Chothia [21] proposed to classify proteins as four basic types or classes according to their α-helix and β-sheet content: “All-α”, “All-β”, “α/β”, and “α+β” classes. The first stage of the two stage method developed by Fadime, O¨zlem and Metin [9] is able to determine the class of unknown proteins with 100% accuracy. In the second stage they use a probabilistic approach based on their stage one results. The amino acid sequences of the training dataset are distributed into overlapping sequence groups of three to seven residues. These groups are used to calculate the probability statistics for secondary structure. Specifically, the secondary structure at a particular sequence location is determined by comparing the probabilities that an amino acid residue is a particular secondary structure type based on the statistics.

Kabsch and Sander developed a set of simple and physically motivated criteria for secondary structure, programmed as a pattern-recognition process of hydrogen-bonded and geometrical features extracted from x-ray coordinates [24]. This DSSP (Define

Secondary Structure of Proteins) algorithm is the standard method for assigning secondary structure to the primary structure (amino acids) of a protein. Depending on the pattern of hydrogen bonds, DSSP recognizes eight types or states of secondary structure. The 3-helix (3/10 helix), alpha helix, and 5 helix (pi helix) are symbolized as G, H and I, respectively. DSSP recognizes two types of hydrogen-bond pairs in beta sheet structures, the parallel and antiparallel bridge. Residue in isolated beta-bridge is symbolized by B, whereas E represents an extended strand, and participates in a beta ladder. The remaining types are T for hydrogen bonded turn, and S for bend. There is also blank or "-" meaning "loop" or "other." These eight types are usually grouped into three classes: helix (G, H, and I), strand/sheet (E and B) and loop/coil (all others).

The work presented herein was influenced by the aforementioned approaches. It also was inspired by the work of Maglia, Leopold, and Ghatti [22] which utilized a data mining approach based on rule induction from coverings in order to identify non-independence in phylogenetic data. Although this appeared to be a promising solution for the phylogenetic data non-independence problem, it suffered from exponential computational complexity (which was in part addressed by a parallelized implementation that was tailored for the phylogenetic data by Leopold et al. [23]), as well as the strictness required for the resulting rules (i.e., all rules had to be correct for all instances in the dataset). A relaxation of that restrictive requirement for the association rules is discussed in Sections "Methods" and "Main RT-RICO Rule-Generation Algorithm"; this modification allowed our research team to discover meaningful rules in another problem domain, protein datasets.

Methods

BLAST-RT-RICO Approach

BLAST-RT-RICO (Relaxed Threshold Rule Induction from Coverings) employs a rule-based data mining approach to predict protein secondary structure. Given an input, protein *A* (where *A* is an amino acid sequence, $A = a_1, a_2, \ldots a_N$), a protein BLAST search (Web-based NCBI/BLAST/BLASTp suite, with PSI-BLAST algorithm) is performed using *A* as the query sequence. BLAST returns a list of proteins with significant sequence alignments. Suitable proteins from this list and related data from the PDB database are chosen to form the training dataset for protein *A*. The RT-RICO algorithm generates rules from the training dataset, and the rules are used to predict the secondary structure for protein *A*. The output is the predicted secondary structure sequence *X*. A flowchart outlining the BLAST-RT-RICO approach is depicted in Fig. 1.

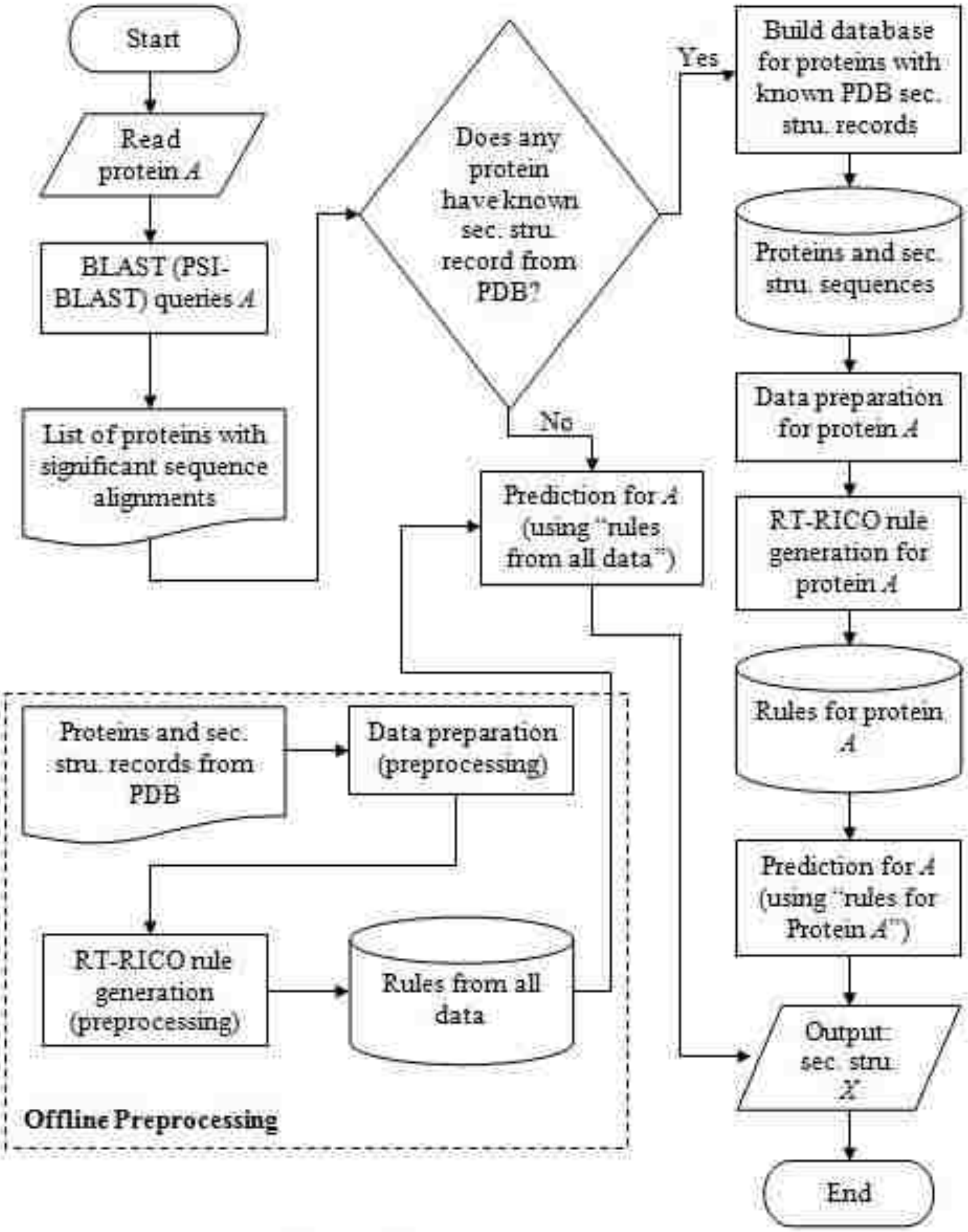

Start
Read protein A
BLAST (PSI-BLAST) queries A
List of proteins with significant sequence alignments
Does any protein have known sec. stru record from PDB?
Yes
No
Build database for proteins with known PDB sec. stru. records
Proteins and sec. stru. sequences
Data preparation for protein A
RT-RICO rule generation for protein A
Rules for protein A
Prediction for A (using "rules for Protein A")
Prediction for A (using "rules from all data")
Proteins and sec. stru. records from PDB
Data preparation (preprocessing)
RT-RICO rule generation (preprocessing)
Rules from all data
Offline Preprocessing
Output: sec. stru X
End

Fig. 1. Flowchart of the BLAST-RT-RICO approach for solving the protein secondary structure prediction problem

In this method, a separate training dataset is constructed for each protein prediction. For example, in the RS126 set (of 126 proteins), it is possible to have 126 different training datasets. The individual training dataset construction and corresponding rule generation operations are performed for each protein. It is important that the training and prediction response time be reasonable for each protein prediction request. In our implementation of the algorithm, each protein request can be completed within minutes, which includes both training time and prediction time. Although the overall prediction time is very reasonable, for future improvements it is useful to identify the bottleneck of the algorithm's performance. Referring to Fig. 1, the most computationally expensive (in terms of time complexity) steps of the algorithm are "RT-RICO rule generation for protein *A*", and "RT-RICO rule generation (preprocessing)."

BLAST-RT-RICO Step 1, Online BLAST and PDB Data Match

As shown in Fig. 1, given as input a protein *A*, $A = a_1, a_2, \ldots a_N$, a BLAST search is performed using *A* as the query sequence. For our implementation, a Web crawler program is used for the BLAST queries. The BLAST search returns a list of proteins with significant sequence alignments and corresponding BLAST scores. Proteins with a score less than or equal to 30 are first removed from the list. The test protein *A* is also removed if it appears in the list, so that it will be excluded from the training dataset. Some of these proteins may have corresponding secondary structure records in the PDB database [10]. A query is made to check if any protein from the list already has a known secondary structure record from the PDB database. If this is the case, then the proteins with corresponding secondary structure records are retrieved; they form the inputs to the next step, data preparation.

If a protein from the list does not have a known secondary structure record in the PDB database, the prediction for that protein needs to be handled slightly differently; it will require data from offline preprocessing, which is discussed in Section "BLAST-RT-RICO, Offline Preprocessing." For the RS126 set, only one protein falls into this category. For the CB396 set, only nine proteins fall into this category. Thus, only a very small percentage of proteins from the test datasets need data from offline preprocessing.

After experimenting with a number of test proteins, we decided to use a BLAST score of 30 as the cutoff in this step in the BLAST-RT-RICO processing; this, in part, was because we found that alignments scores less than 30 did not improve the prediction. However, there is no evidence that 30 is the best choice. We intend to further investigate how the selection of the E-value affects the final Q_3 prediction accuracy. It should be noted that, although considered a good indicator of the alignment, the BLAST E-value only describes the likelihood that a sequence with a similar score will occur in the database by chance.

BLAST-RT-RICO Step 2, Data Preparation

The proteins with significant sequence alignments and corresponding secondary structure records are inputs to the data preparation step. For test protein *A*, there is a set of protein primary structure sequence B_i and a set of corresponding secondary structure sequence C_i where $B_i \in \{B_1, B_2, B_3, B_4, \ldots B_y\}, C_i \in \{C_1, C_2, C_3, C_4, \ldots C_y\},$ the protein primary structure sequence is $B_i = b_{i,1}, b_{i,2}, b_{i,3}, \ldots b_{i,n_i}$ and the corresponding secondary structure sequence is $C_i = c_{i,1}, c_{i,2}, c_{i,3}, \ldots c_{i,n_i}$ Sequences B_1 to B_y are not necessarily of the same length, because they represent different proteins; in other words, sequence *i* has length n_i. Here each $b_{i,j}$ is an element of a set of 20 amino acids, $\{A,R,N\ldots V\}$. Initially, $c_{i,j}$ is an element of a set of eight-state secondary structures, *{H, G, I, E, B, T, S, -}*, as represented in the PDB database. It is then converted to an element of a set of four-state secondary structures, *{H, E, C, -}*.

The protein secondary structure sequences from PDB are formed by elements of eight states of secondary structure, {H, G, I, E, B, T, S, -}. To facilitate rule generation those eight states were converted to four states as follows:

(G, H, I) => Helix H

(E, B) => Sheet E

(T, S) => Coil C

(-) => “-”

Whereas rule generation uses a four-state “decision” attribute, the final Q_3 score calculation uses a three-state attribute where:

(G, H, I) => Helix H

(E, B) => Sheet E

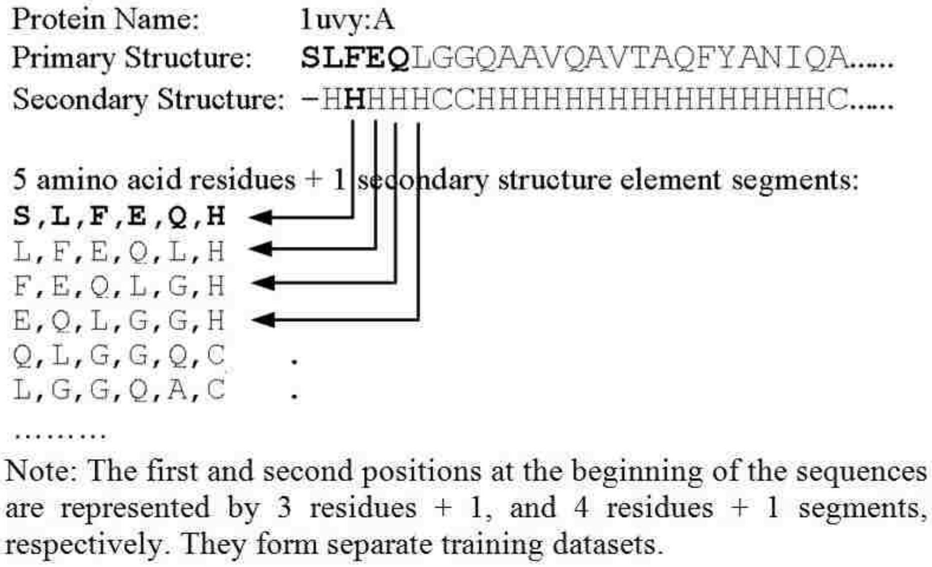
(Rest) => Coil C

A four-state (rather than three-state) decision attribute is used for rule generation, because the chemical structures of the secondary structure elements can be closely grouped into four types (Helix H, Sheet E, Coil C and "-") in general. As a result, a four-state decision attribute allows more meaningful rules to be generated, and hence improves prediction accuracy (as compared to a three-state decision attribute). Because the standard Q_3 score uses a three-state attribute, a simple conversion is done before the final Q_3 score calculation.

The basis for the rule-based approach is to first search segments of amino acid sequences of known protein secondary structures, and then find the rules that relate amino acid residues to secondary structure elements. The generated rules are subsequently used to predict the secondary structure. Klepeis and Floudas [24] showed that the use of overlapping segments of five residues is very effective in predicting the helical segments of proteins. Thus, the overlapping 5-residue segments approach was used to prepare the training data records. As shown in Fig. 2, for each secondary structure element, five "neighboring" amino acid residues were extracted to form a segment of five amino acid residues, plus one secondary structure element. These segments were used as input to the RT-RICO rule generation algorithm to produce rules.

If B_i is the primary structure sequence, C_i is the secondary structure sequence (as shown in Fig. 2), and the length of the sequence(s) is n_i, then each 5-residue segment is of the form: $b_{i,j-2}$, $b_{i,j-1}$, $b_{i,j}$, $b_{i,j+1}$, $b_{i,j+2}$, $c_{i,j}$; and j has a value from 3 to $(n_i - 2)$. This data preparation step is performed for all B_i and C_i pairs, where i is from 1 to y.

The 5-residue segments are inputs to the RT-RICO rule generation algorithm. They are represented as 6-tuples, where the first five elements of a 6-tuple are formed by amino acid residues, {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}, and the last element of a 6-tuple is formed by one of four secondary structure states {H, E, C, -}. The last element is considered the "decision" (or determination) attribute. In other words, the input to BLAST-RT-RICO Step 3, rule generation, is in the form of an $m \times (n+1)$ matrix, where m is the number of all entities (the number of 5-residue plus one secondary structure element segments), and $n = |S|$ (the number of attributes, where $n = 5$ in this case). It should be noted that Fig. 2 only shows the extraction of 5-residue segments from one protein record (B_i *and* C_i); this extraction process actually is performed for all protein records (all B_i and C_i pairs, where *i* is from *1* to *y*, and sequences B_1 to B_y are not necessarily of the same length).



Protein Name: 1uvy:A
Primary Structure: **SLFEQ**LGGQAAVQAVTAQFYANIQA......
Secondary Structure: –H**H**HHHCCHHHHHHHHHHHHHHHHC......

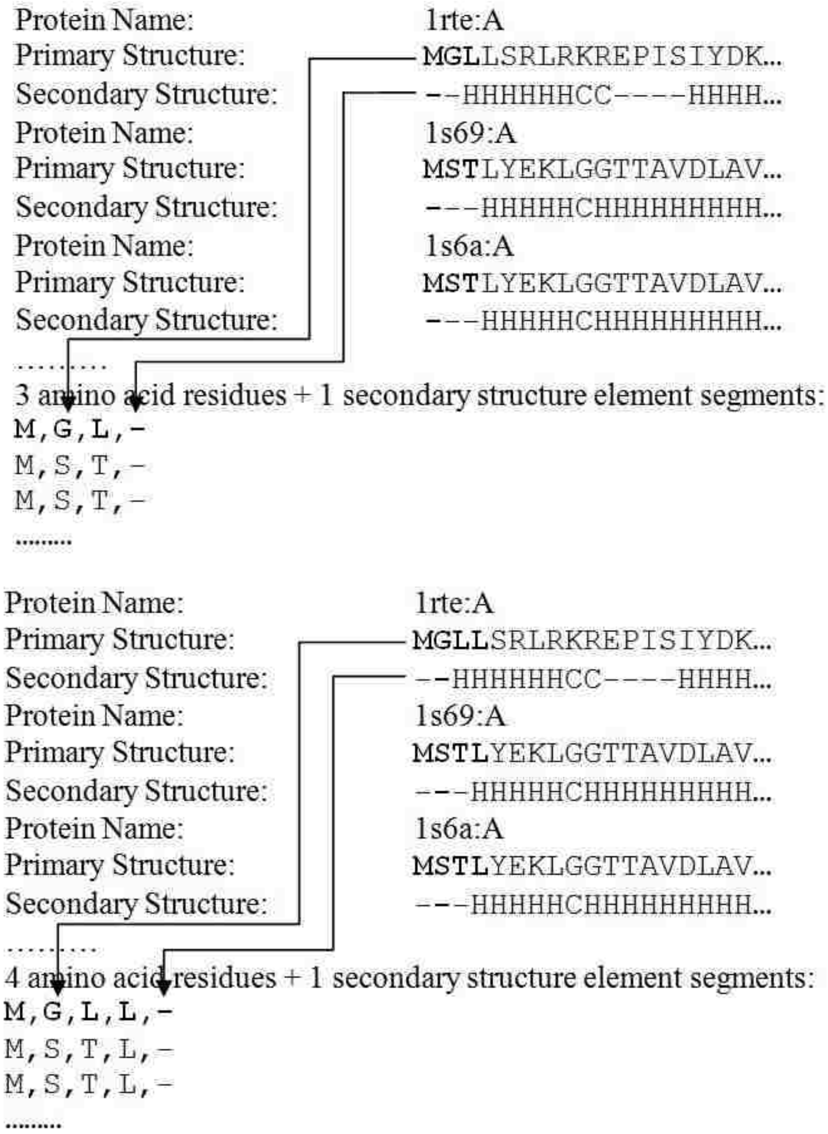
5 amino acid residues + 1 secondary structure element segments:
S,L,F,E,Q,H
L,F,E,Q,L,H
F,E,Q,L,G,H
E,Q,L,G,G,H
Q,L,G,G,Q,C .
L,G,G,Q,A,C .
....

Note: The first and second positions at the beginning of the sequences are represented by 3 residues + 1, and 4 residues + 1 segments, respectively. They form separate training datasets.

Fig. 2. Protein primary structure 5-residue segments and related secondary structure elements representation.

As shown in Fig. 2, for a protein amino acid sequence and corresponding secondary structure sequence of length *k* (say $k = n_i$), only the secondary structure elements from the third position to position *(k-2)* are extracted for the 5-residue segments. The first and second positions at the beginning of the secondary structure sequence, as well as the last and second-to-last positions at the end of the secondary structure sequence, are not extracted as 5-residue segments. To handle these positions, extractions are done slightly differently, as shown in Fig. 3.

These 3-residue and 4-residue segments are used as input to the RT-RICO rule generation algorithm (as introduced in Section “BLAST-RT-RICO Step 3, Rule Generation”, with more details given in Section “Main RT-RICO Rule-Generation Algorithm”) to generate rules. The input to RT-RICO step 3, Rule Generation, is also in the form of an $m \times (n+1)$ matrix, where $n = 3$ for 3-residue segments, and $n=4$ for 4-residue segments. The same rule generation algorithm applies to all of these segments. The rules generated subsequently are used in step 4 to predict the secondary structure elements at the first and second positions, as well as the last and second-to-last positions of unknown secondary structure sequences, respectively.


Protein Name: 1rte:A
Primary Structure: MGLLSRLRKREPISIYDK...
Secondary Structure: --HHHHHHCC----HHHH...
Protein Name: 1s69:A
Primary Structure: MSTLYEKLGGTTAVDLAV...
Secondary Structure: ---HHHHHCHHHHHHHHH...
Protein Name: 1s6a:A
Primary Structure: MSTLYEKLGGTTAVDLAV...
Secondary Structure: ---HHHHHCHHHHHHHHH...
....
3 amino acid residues + 1 secondary structure element segments:
M,G,L,-
M,S,T,-
M,S,T,-
....
Protein Name: 1rte:A
Primary Structure: MGLLSRLRKREPISIYDK...
Secondary Structure: --HHHHHHCC----HHHH...
Protein Name: 1s69:A
Primary Structure: MSTLYEKLGGTTAVDLAV...
Secondary Structure: ---HHHHHCHHHHHHHHH...
Protein Name: 1s6a:A
Primary Structure: MSTLYEKLGGTTAVDLAV...
Secondary Structure: ---HHHHHCHHHHHHHHH...
....
4 amino acid residues + 1 secondary structure element segments:
M,G,L,L,-
M,S,T,L,-
M,S,T,L,-
....

Note: The last and second-to-last positions at the end of the sequences are also represented by 3 residues + 1, and 4 residues + 1 segments, respectively. The segments are generated in a similar way, but form separate training datasets.

Fig. 3. Protein primary structure 3-residue segments and related secondary structure elements representation, protein primary structure 4-residue segments and related secondary structure elements representation, at the beginning of the sequences.

For an amino acid sequence of length *k*, *(k-4)* 5-residue segments are extracted, whereas only two 3-residue segments (in the first and last positions), and two 4-residue segments (in the second and second-to-last positions) are extracted. As the extraction was done for a large number of proteins, the rule generation and prediction operations in later steps involved mostly 5-residue segments in terms of the training data size. For this reason, only 5-residue segment numbers were recorded in the prediction result tables, and only 5-residue segment numbers were considered in the algorithm time complexity that is discussed in later sections.

BLAST-RT-RICO Step 3, Rule Generation

RT-RICO generates rules based on the segments in the form of an $m \times (n+1)$ matrix. The main RT-RICO rule generation algorithm is explained in Section “Main RT-RICO Rule-Generation Algorithm.” Some examples of the generated rules are shown in Fig. 4 in two separate formats. The first format is intended to be read by the computer programs at the later prediction stage (i.e., the computer rule format). The second format is intended to be read by the user (i.e., the human rule format). The first rule (in human rule format) is interpreted as follows: if the fourth position attribute (or “3” as interpreted by program) is “C”, and the fifth position attribute (or “4” as interpreted by program) is “C”, then the sixth attribute (decision attribute, or “5” as interpreted by program) is “H” with a confidence of 91.53% and a support of 0.04864442%. The definitions of confidence and support can be found in [26].

The corresponding first rule (in computer rule format) is interpreted as follows: if the first position attribute is “+” (representing any amino acid element), the second position attribute is “+”, the third position attribute is “+”, the fourth position attribute is “C”, and the fifth position attribute is “C”, then the sixth attribute (i.e., the decision attribute) is “H.” The number of occurrences of the fourth position attribute (which is “C”) and the fifth position attribute (which is “C”) equals 720 among all inputs to RT-RICO. The number of occurrences of the fourth position attribute (which is “C”), the fifth position attribute (which is “C”), and the sixth attribute (which is “H”), equals 659 among all inputs to RT-RICO. The confidence is 91.53% and the support is 0.04864442%.

```
+,+,+,C,C,H,91.53,720,659,0.04864442
+,+,C,C,+,H,91.69,722,662,0.04886586
+,+,A,C,Y,H,100.00,26,26,0.00191920
……
(3,C)(4,C) -> (5, H), 91.53%,
occurrences of ((3,C)(4,C)) = 720,
occurrences of ((3,C)(4,C) -> (5, H)) = 659, Support
% = 0.04864442
(2,C)(3,C) -> (5, H), 91.69%,
occurrences of ((2,C)(3,C)) = 722,
occurrences of ((2,C)(3,C) -> (5, H)) = 662, Support
% = 0.04886586
(2,A)(3,C)(4,Y) -> (5, H), 100.00%,
occurrences of ((2,A)(3,C)(4,Y)) = 26,  occurrences
of ((2,A)(3,C)(4,Y) -> (5, H)) = 26, Support % =
0.00191920
……
```

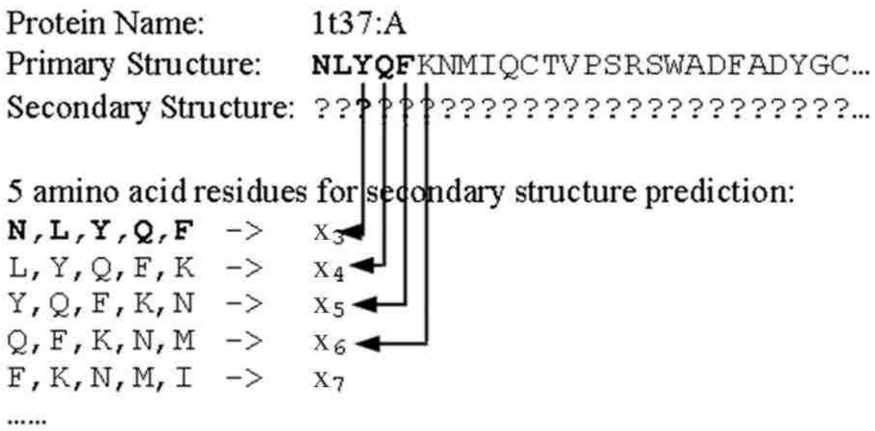
Fig. 4. Sample rules generated by RT-RICO.

BLAST-RT-RICO Step 4, Prediction

Finally RT-RICO loads protein primary structures from the test dataset (a single protein *A* for this case), and predicts the secondary structure elements.

As shown in Fig. 5, for each secondary structure element prediction position (for a corresponding amino acid sequence of length *k*, from position *3* to *k-2*), five “neighboring” amino acid residues are extracted to form a segment of five amino acid residues. Each of these segments is compared with the generated rules (generated from 5-residue segments). If a segment matches a rule, the support value of the rule is taken into consideration for the prediction of the related secondary structure element.

The algorithm first searches for matching rules with 100% confidence value. The secondary structure element with the highest total support value (among 100% confidence value rules) is selected. If no matching rule exists among 100% confidence value rules, the algorithm then searches for other matching rules (with confidence values greater than or equal to 90%, but less than 100%). The secondary structure element with the highest total support value among these rules is selected as the predicted secondary structure element for that specific position. If no matching rule is found for the segment at all, the secondary structure of the previous position is used as the predicted secondary structure.



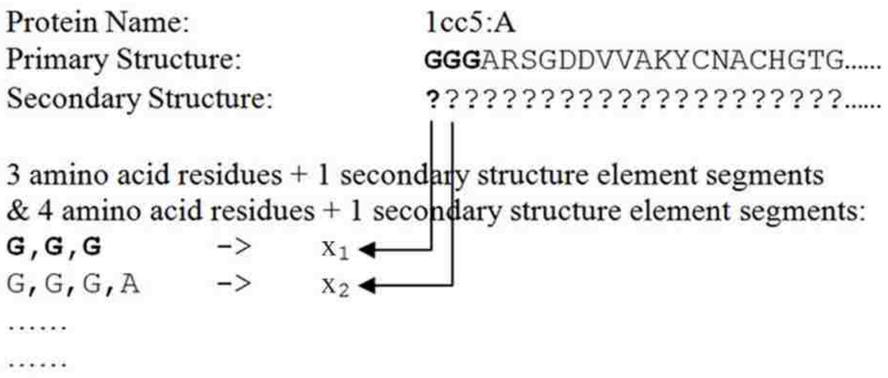
```
Protein Name:           1t37:A
Primary Structure:      NLYQFKNMIQCTVPSRSWADFADYGC...
Secondary Structure:    ?????????????????????????...

5 amino acid residues for secondary structure prediction:
N,L,Y,Q,F  ->   x3
L,Y,Q,F,K  ->   x4
Y,Q,F,K,N  ->   x5
Q,F,K,N,M  ->   x6
F,K,N,M,I  ->   x7
......
```

Fig. 5. Protein primary structure 5-residue segments and related secondary structure elements prediction. Here m_i is an element of the set {H,E,C,-}. It is then converted to an element of the set {H, E, C}. Note: The first and second positions at the beginning of the sequence are represented (i.e., predicted) by 3-residue, and 4-residue segments, respectively.

To predict the first and second positions at the beginning of a secondary structure sequence, and the last and second-to-last positions at the end of a secondary structure sequence, three or four "neighboring" amino acid residues are extracted, as shown in Fig. 6. The same prediction algorithm mentioned above is responsible for the secondary structure prediction at these positions, but instead using rules generated from 3-residue and 4-residue segments as was discussed in Section "BLAST-RT-RICO Step 2, Data Preparation."

The output of the prediction is a sequence of secondary structure elements $X = x_1, x_2, \ldots x_N$ where each x_i is an element of a set of four-state secondary structures, *{H,E,C,-}*. The Q_3 score calculation uses a three-state decision attribute. Hence x_i is first converted to an element of a set of three-state secondary structure, *{H,E,C}*, before the final Q_3 score calculation.



Protein Name: 1cc5:A

Primary Structure: **GGG**ARSGDDVVAKYCNACHGTG......

Secondary Structure: **?**?????????????????????......

3 amino acid residues + 1 secondary structure element segments
& 4 amino acid residues + 1 secondary structure element segments:

G,G,G -> x_1

G,G,G,A -> x_2

......

......

Fig. 6. Protein primary structure 3-residue, 4-residue segments, and related secondary structure elements prediction. Here m_i is an element of the set {H,E,C,-}. It is then converted to an element of the set {H, E, C}. Note: The last and second-to-last positions at the end of the sequence are also represented (i.e., predicted) by 3-residue, and 4-residue segments, respectively.

BLAST-RT-RICO, Offline Preprocessing

As previously mentioned, if no protein with significant sequence alignments has corresponding known secondary structure sequence from the PDB database (i.e., the answer is “no” in Fig. 1.), the prediction for test protein *A* needs to be handled in a slightly different manner. Some proteins and secondary structures need to be selected to generate rules for the prediction of protein *A*. The operations can be performed offline primarily because it is not necessary to perform rule generation for every protein prediction; instead, rules can be generated once and used for all the proteins that fall into this category.

In offline preprocessing, all proteins and corresponding secondary structure sequences from the PDB database are downloaded to form an initial dataset. Proteins from the test datasets (RS126 or CB396) are first removed from this dataset, so that they will be excluded from the possible training datasets. Protein domains from different protein families are selected to form the training datasets; see Table 2 for the number of protein domains in each training dataset for the RS126 and CB396 test datasets. The reason for having two different training datasets is primarily due to the fact that the

RS126 and CB396 data need to first be removed from the initial dataset. If a new system is to be constructed to predict previously unknown proteins, a single training dataset will be sufficient for offline preprocessing.

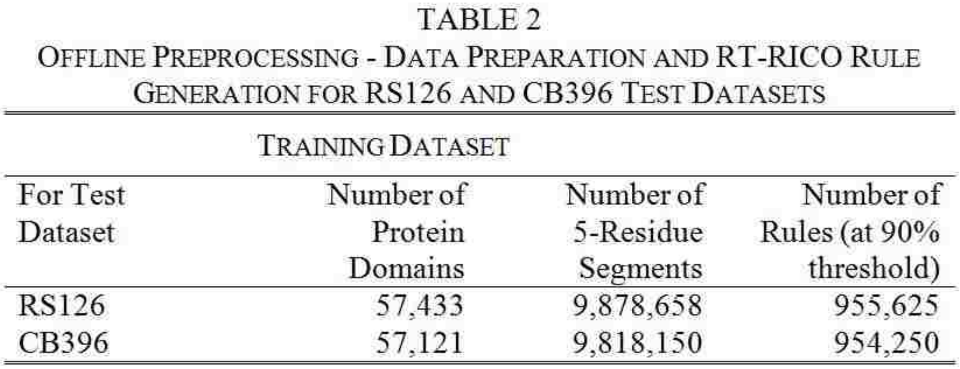


TABLE 2
OFFLINE PREPROCESSING - DATA PREPARATION AND RT-RICO RULE GENERATION FOR RS126 AND CB396 TEST DATASETS

	TRAINING DATASET		
For Test Dataset	Number of Protein Domains	Number of 5-Residue Segments	Number of Rules (at 90% threshold)
RS126	57,433	9,878,658	955,625
CB396	57,121	9,818,150	954,250

After the selection of the protein domains, we have a set of protein primary structure sequence B_i and corresponding secondary structure sequence C_i where $B_i \in \{B_1, B_2, B_3, B_4, \ldots B_z\}$ and $C_i \in \{C_1, C_2, C_3, C_4, \ldots C_z\}$. The data preparation step for offline preprocessing is the same as the data preparation step earlier described in Section “BLAST-RT-RICO Step 2, Data Preparation.” As shown in Fig. 2, for each secondary structure element, five “neighboring” amino acid residues are extracted to form a segment of five amino acid residues, plus one secondary structure element. Fig. 3 shows how the beginning and the end of the sequences are handled. These segments are used as input to the RT-RICO rule generation algorithm to generate association rules. The rule generation step for offline preprocessing is the same as the rule generation step described in Section “BLAST-RT-RICO Step 3, Rule Generation.”

Finally, the primary structure sequence of protein *A* is loaded, and the secondary structure elements are predicted using the rules generated from offline preprocessing (rules from all data). The prediction step here is the same as the prediction step described in Section “BLAST-RT-RICO Step 4, Prediction” above.

Main RT-RICO Rule-Generation Algorithm

Although the RT-RICO protein secondary structure prediction method consists of the steps mentioned in Section “Methods”, the most computationally intensive part is in the rule generation, performed both in the third step and during offline preprocessing.

Rule Induction From Coverings

RT-RICO is based on a previously implemented method called RICO (Rule Induction from Coverings) [22]. RICO uses some of the concepts introduced by Pawlak [27] for rough sets, a classification scheme based on partitions of entities in a dataset [28]. In this approach, if *S* is a set of attributes and *R* is a set of decision attributes (i.e., attributes whose values we are interested in being able to determine if the values of the attributes in the set *S* are known), then a covering *P* of *R* in *S* can be found if the following three conditions are satisfied:

i. *P* is a subset of *S*.

ii. *R depends* on *P* (i.e., *P determines R*). That is, if a pair of entities *x* and *y* cannot be distinguished by means of attributes from *P*, then *x* and *y* also cannot be distinguished by means of attributes from *R*. If this is true, then entities *x* and *y* are said to be *indiscernible* by *P* (and, hence, *R*), denoted $x \sim_P y$. An *indiscernibility relation* $\sim_P$ is such a partition over all entities in the data set.

iii. *P* is minimal.

Condition (ii) is true if and only if an equivalent condition ≤, known as the *attribute dependency inequality*, holds for *P** and *R**, the partitions of all attributes and decisions generated by *P* and *R*, respectively, where, for a set of attributes *A*:

$$A^* = \prod_{a \in A} \sim [a]^*$$

The inequality $P^* \leq R^*$ holds if and only if for each block *B* of *P**, there exists a block *B′* of *R** such that *B* is a subset of *B′*.

Once a covering is found, it is a straightforward process to induce rules from it. For example, if a set of attributes $P = \{a_1, a_2\}$ is found to determine a set of attributes $R = \{a_3\}$ (i.e., *P* is a covering for *R*), then rules of the form $(a_1, v_{1)} \wedge (a_2, v_2) \rightarrow (a_3, v_3)$ (read as “if a_1 equals v_1 and a_2 equals v_2, then a_3 equals v_3) can be generated where v_1, v_2, and v_3 are actual values of attributes a_1, a_2, and a_3, respectively, for which the relationship holds in the dataset. Such a rule also conveys a notion of non-independence between the attributes in the sets *P* and *R* (e.g., a_3 is not independent of a_1 and a_2). Here non-independence means that the relationship between the two attributes could be correlation, dependency, or co-dependency.

Relaxed Attribute Dependency Inequality

All rules generated from coverings in this manner are “perfect” in the sense that there is no instance in the dataset for which the rule is not true. In order to relax this restriction somewhat (in much the same way that rules generated by decision tree induction are not always true for all instances in the dataset), the definition of the attribute dependency inequality can be modified as follows.

Definition 1: Relaxed Attribute Dependency Inequality

The inequality $P^* \leq_r R^*$ holds if and only if *there exists* a block *B* of *P**, and *there exists* a block *B′* of *R** such that *B* is a subset of *B′*.

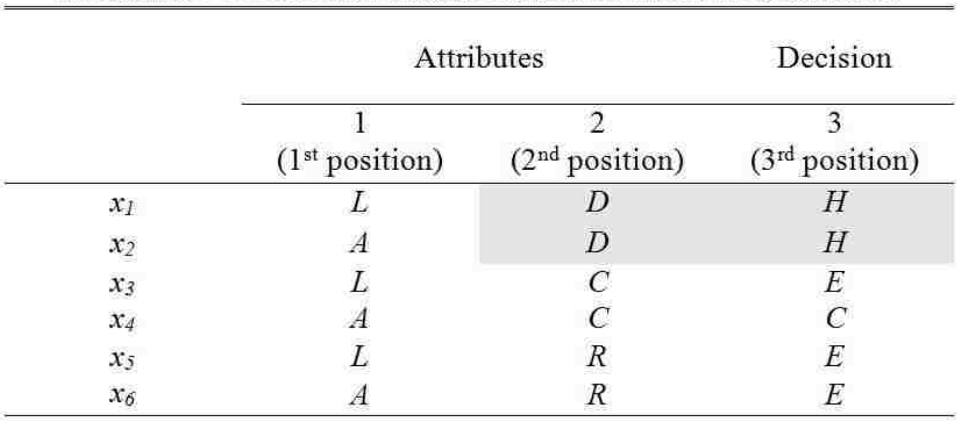
As an example for the dataset of Table 3, let $P = \{2\}$ and $R = \{3\}$. Then

$\{2\}^* = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6\}\}$

$\{3\}^* = \{\{x_1, x_2\}, \{x_3, x_5, x_6\}, \{x_4\}\}$

There exists a block $B = \{x_1, x_2\}$ in $\{2\}^*$ and a block $B' = \{x_1, x_2\}$ in $\{3\}^*$ such that $B \subseteq B'$. Thus, $\{2\}^* \leq_r \{3\}^*$ which means that $\{3\}$ depends on $\{2\}$ (i.e., $\{2\} \rightarrow_r \{3\}$) for at least *some* values of $\{2\}$. More specific rules can then be deduced from this relationship, such as $(2, D) \rightarrow (3, H)$.

TABLE 3
A DECISION TABLE FOR INDISCERNIBLE RELATIONSHIPS EXAMPLE



	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*L*	*D*	*H*
x_2	*A*	*D*	*H*
x_3	*L*	*C*	*E*
x_4	*A*	*C*	*C*
x_5	*L*	*R*	*E*
x_6	*A*	*R*	*E*

Relaxed Coverings

Similarly, the definition of a covering can be relaxed in order to induce rules depending on as small a number of attributes as possible.

Definition 2: Relaxed Coverings

A subset *P* of the set *S* is called a *relaxed covering* of *R* in *S* if and only if $P \rightarrow_r R$ and *P* is minimal in *S*. This is equivalent to saying that a subset *P* of the set *S* is a *relaxed covering* of *R* in *S* if and only if $P \rightarrow_r R$ and no proper subset *P'* of *P* exists such that $P' \rightarrow_r R$.

As an example for the dataset of Table 3, suppose rules need to be induced for $R = \{3\}$. The covering *{1, 2}* can be used; that is, for any assignment of values for the covering *{1, 2}*, each entity in Table 3 will induce a rule for *{3}*. But, instead of inducing a rule by looking at combinations of values for *{1, 2}*, such as $(1, L) \wedge (2, D) \rightarrow (3, H)$, rules are to be induced based on values for only *{1}* or *{2}*. Thus, $(2, D) \rightarrow (3, H)$ will be generated as a rule since $\{2\} \rightarrow_r \{3\}$ and *{2}* is minimal in *{1, 2}*. In this manner, *{2}* is a *relaxed covering* of *{3}*.

Checking Attribute Dependency

To implement rule induction from coverings with the relaxed constraints, it is necessary to use the concept of checking attribute dependency, which was introduced by Grzymala-Busse [28]. In order for *P* to be a relaxed covering of *R* in *S*, the following conditions must be true:

i. *P* must be a subset of *S*,

ii. *R* must depend on set *P* (for some values of *P*), and

iii. *P* must be minimal.

For the specific application of generating rules for protein secondary structure prediction, rules involving more attributes are preferred over rules involving fewer attributes, because they typically generate higher confidence values. In addition, all the possible attribute position combinations are needed to predict secondary structure. As a result, condition (iii) is not enforced for rule generation in our implementation. In fact, condition (iii) cannot be enforced for this particular application; otherwise, many meaningful rules involving multiple attributes and high confidence values would not be generated, leading to inaccurate predictions.

Condition (ii) is true if and only if the relaxed attribute dependency inequality, $P^* \leq_r R^*$, is satisfied. The question then becomes how this inequality can be checked efficiently. For each set P, a new partition U, generated by P, must be determined. For partitions π and τ of U, $\pi \cdot \tau$ is a partition of U such that two entities, x and y, are in the same block of $\pi \cdot \tau$ if and only if x and y are in the same block for both partitions π and τ of U. For example, referring to Table 4,

$\{1\}^* = \{\{x_1, x_2, x_5, x_6\}, \{x_3, x_4\}\}$

$\{2\}^* = \{\{x_1, x_2, x_4, x_5\}, \{x_3, x_6\}\}$

$\{1\}^* \cdot \{2\}^* = \{\{x_1, x_2, x_5\}, \{x_3\}, \{x_4\}, \{x_6\}\}$

That is, for $\{1\}^*$ and $\{2\}^*$, two entities x_1 and x_2 are in the same block of $\{1\}^* \cdot \{2\}^*$ if and only if x_1 and x_2 are in the same block of $\{1\}^*$ and in the same block of $\{2\}^*$. Further, the relaxed covering of $\{3\}$ is $\{1, 2\}$, because $\{1\}^* \cdot \{2\}^* \leq_r \{3\}^*$, and $\{1, 2\}$ is minimal since $\{1\}^* \leq_r \{3\}^*$ and $\{2\}^* \leq_r \{3\}^*$ are not true.

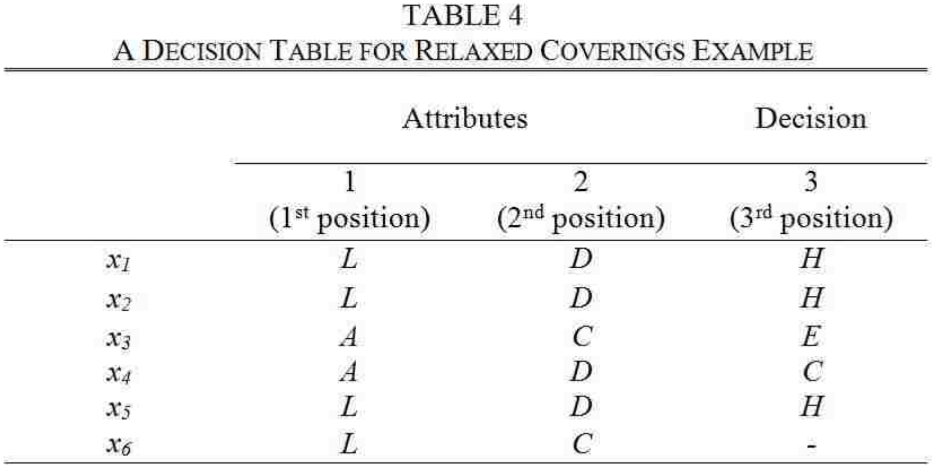


TABLE 4
A DECISION TABLE FOR RELAXED COVERINGS EXAMPLE

	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*L*	*D*	*H*
x_2	*L*	*D*	*H*
x_3	*A*	*C*	*E*
x_4	*A*	*D*	*C*
x_5	*L*	*D*	*H*
x_6	*L*	*C*	-

Finding the Set of All Relaxed Coverings

The algorithm R-RICO (Relaxed Rule Induction from Coverings) which is given below can be used to find the set C of all relaxed coverings of R in S (as well as the related rules).

Let *S* be the set of all attributes, and let *R* be the set of all decision attributes. Let *k* be a positive integer. The set of all subsets of the same cardinality *k* of the set *S* is denoted $P_k = \{\{x_{i1}, x_{i2}, \ldots , x_{ik}\} \mid x_{i1}, x_{i2}, \ldots , x_{ik} \in S\}$ [28].

Algorithm 1: R-RICO

```
begin
    for each attribute x in S do
        compute [x]*;
    compute partition R*
    k:=1
    while k ≤ |S| do
        for each set P in $P_k$ do
            if ($\prod_{x \in P}$ [x]* $\leq_r$ R*) then
            begin
                find the attribute values from the first block B of P
                and from the first block B’ of R;
                add the rule to the output file;
            end
        k := k+1;
    end-while
end-algorithm.
```

Note that the condition (iii) for a relaxed covering is not enforced in the R-RICO algorithm.

The time complexity of the R-RICO algorithm is exponential to |*S*|, the number of attributes in the dataset.

RT-RICO Algorithm

The R-RICO algorithm produces rules that are 100% correct. However, unlike decision tree induction, R-RICO produces a more comprehensive rule set. The algorithm can be further modified to satisfy some particular level of uncertainty in the rules (e.g., the rule is ≥ 50% true). That is, rather than just reporting a rule *R*, the rule can be reported as a tuple *(R, p)* where *p* is the probability that rule *R* is true. To accommodate this information in the rules, the definition of attribute dependency inequality must be further modified as in Definition 3.

Definition 3: Relaxed Attribute Dependency Inequality with Threshold

Set *R* depends on a set *P* with threshold probability *t* ($0 < t \leq 1$), and is denoted by $P \rightarrow_{r,t} R$ if and only if $P^* \leq_{r,t} R^*$ and there exists a block *B* of *P**, and there exists a block *B'* of *R** such that $(|B \cap B'| / |B|) \geq t$.

It can be observed that, when $t=1$, Definitions 1 and 3 represent the same mathematical relation.

As an example, for the dataset of Table 5, let $P = \{1, 2\}$, $R = \{3\}$, and $t = 0.6$. Then the following partitions can be formed:

$\{1\}^* = \{\{x_1, x_6\}, \{x_2, x_3, x_4, x_5\}\}$

$\{2\}^* = \{\{x_1, x_2, x_3, x_4, x_5, x_6\}\}$

$P^* = \{1,2\}^* = \{1\}^* \cdot \{2\}^* = \{\{x_1, x_6\}, \{x_2, x_3, x_4, x_5\}\}$

$R^* = \{3\}^* = \{\{x_1, x_5\}, \{x_2, x_3, x_4, x_6\}\}$

There exists a block $B = \{x_2, x_3, x_4, x_5\}$ in $\{1, 2\}^*$, and there exists a block $B' = \{x_2, x_3, x_4, x_6\}$ in $\{3\}^*$ such that $(|B \cap B'| / |B|) = |\{x_2, x_3, x_4\}| / |\{x_2, x_3, x_4, x_6\}| = 0.75 \geq 0.6$. Thus, $P^* = \{1, 2\}^* \leq_{r,t} R^* = \{3\}^*$, and $\{3\}$ depends on $\{1, 2\}$ (i.e., $\{1, 2\} \rightarrow_{r,t} \{3\}$), with threshold probability *0.6*.

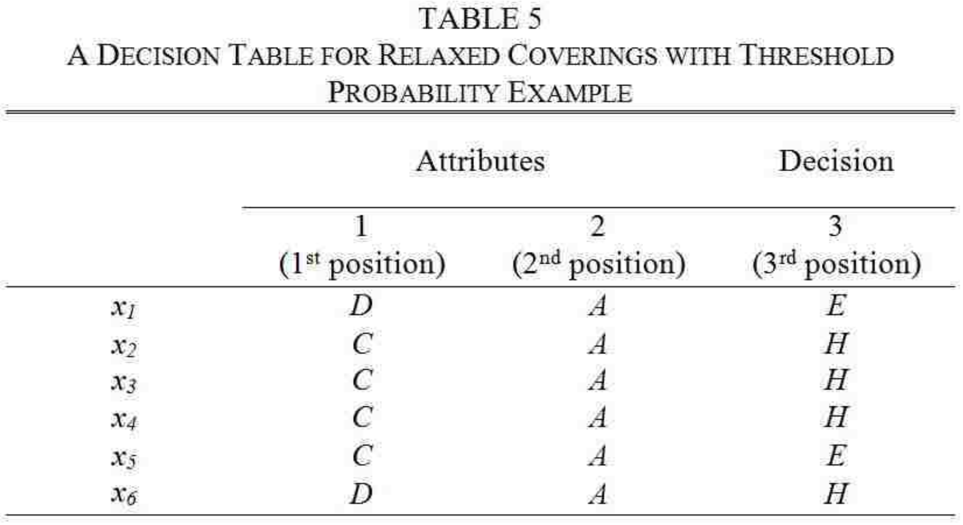


TABLE 5
A DECISION TABLE FOR RELAXED COVERINGS WITH THRESHOLD PROBABILITY EXAMPLE

	Attributes		Decision
	1 (1st position)	2 (2nd position)	3 (3rd position)
x_1	*D*	*A*	*E*
x_2	*C*	*A*	*H*
x_3	*C*	*A*	*H*
x_4	*C*	*A*	*H*
x_5	*C*	*A*	*E*
x_6	*D*	*A*	*H*

The corresponding values of attributes can be found from entities that are in the $B \cap B' = \{x_2, x_3, x_4\}$ for the sets $P = \{1, 2\}$ and $R = \{3\}$; namely, the value of attribute 1 is C, the value of attribute 2 is A at $\{x_2, x_3, x_4\}$, and the value of decision 3 is H for entities $\{x_2, x_3, x_4\}$. The rule induced from $\{1, 2\} \rightarrow_{r,t} \{3\}$ is then $(1, C) \wedge (2, A) \rightarrow (3, H)$ with a probability (confidence) of 75%. Another way to look at this is to note that the number of occurrences of $((1,C)(2,A)) = 4$, and the number of occurrences of $((1,C)(2,A) \rightarrow (3, H)) = 3$.

The definition of relaxed coverings must also be modified to incorporate the notion of the threshold probability as in Definition 4.

Definition 4: Relaxed Coverings with Threshold Probability

Let S be a nonempty subset of a set of all attributes, and let R be a nonempty subset of decision attributes, where S and R are disjoint. A subset P of the set S is called a relaxed covering of R in S with threshold probability t $(0 < t \leq 1)$ if and only if $P \rightarrow_{r,t} R$ and P is minimal in S.

Algorithm RT-RICO (Relaxed Threshold Rule Induction From Coverings) finds the set C of all relaxed coverings of R in S (and the related rules), with threshold probability t $(0 < t \leq 1)$, where S is the set of all attributes, and R is the set of all decisions. The set of all subsets of the same cardinality k of the set S is denoted $P_k = \{\{x_{i1}, x_{i2}, \dots, x_{ik}\} \mid x_{i1}, x_{i2}, \dots, x_{ik} \in S\}$.

Algorithm 2: RT-RICO

```
begin
    for each attribute x in S do
        compute [x]*;
    compute partition R*
    k:=1
    while k ≤ |S| do
        for each set P in $P_k$ do
        if ($\prod_{x \in P}$ [x]* $\leq_{r,t}$ R*) then
        begin
```

```
            find values of attributes from the entities that are in the ($B \cap B'$) such that ($|B \cap B'| / |B|) \geq t$;
            add the rule to the output file;
        end
        k := k+1
    end-while;
end-algorithm.
```

Note that the condition “*P* is minimal in *S*” of a relaxed covering with threshold probability is not enforced in the RT-RICO algorithm. The reason for not implementing this condition is the same as the reason mentioned for the R-RICO algorithm. To generate rules for protein secondary structure prediction, rules involving more attributes are preferred over rules involving fewer attributes, because they typically generate higher confidence values. Also, all the possible attribute position combinations are needed for accurate prediction.

The time complexity of the RT-RICO algorithm is again exponential to $|S|$, the number of attributes in the dataset. The time complexity is in fact $O(m^2 2^n)$, where m is the number of all entities (the number of 5-residue segments), and $n = |S|$ (the number of attributes). It would appear that 2^n dominates the time complexity. But, for the training datasets used for protein secondary structure prediction, $n = |S| = 5$, and m is sufficiently large. Hence, m^2 dominates the time complexity in this case.

As discussed in Section “Methods”, the rules generated by the RT-RICO algorithm are then compared with the proteins in the test dataset to predict the secondary structure elements.

Results

The RS126 set [4] and the CB396 set [2] are both non-redundant test datasets created with the objective of comparing different protein secondary structure prediction methods; it should be noted that the CB396 set does not include protein domains from the RS126 set. As previously mentioned in the Section “Background”, the two datasets have been used as standard datasets to evaluate other prediction methods, and hence were deemed appropriate for evaluating the performance of the RT-RICO protein secondary prediction method.

Table 2 lists the number of protein domains, segments, and rules in the training datasets for offline preprocessing. Table 6 shows a summary of the number of proteins, segments, and rules in each training dataset (the results of BLAST and subsequent operations) for individual proteins; it also shows the performance of the BLAST-RT-RICO method on the RS126 and CB396 test datasets.

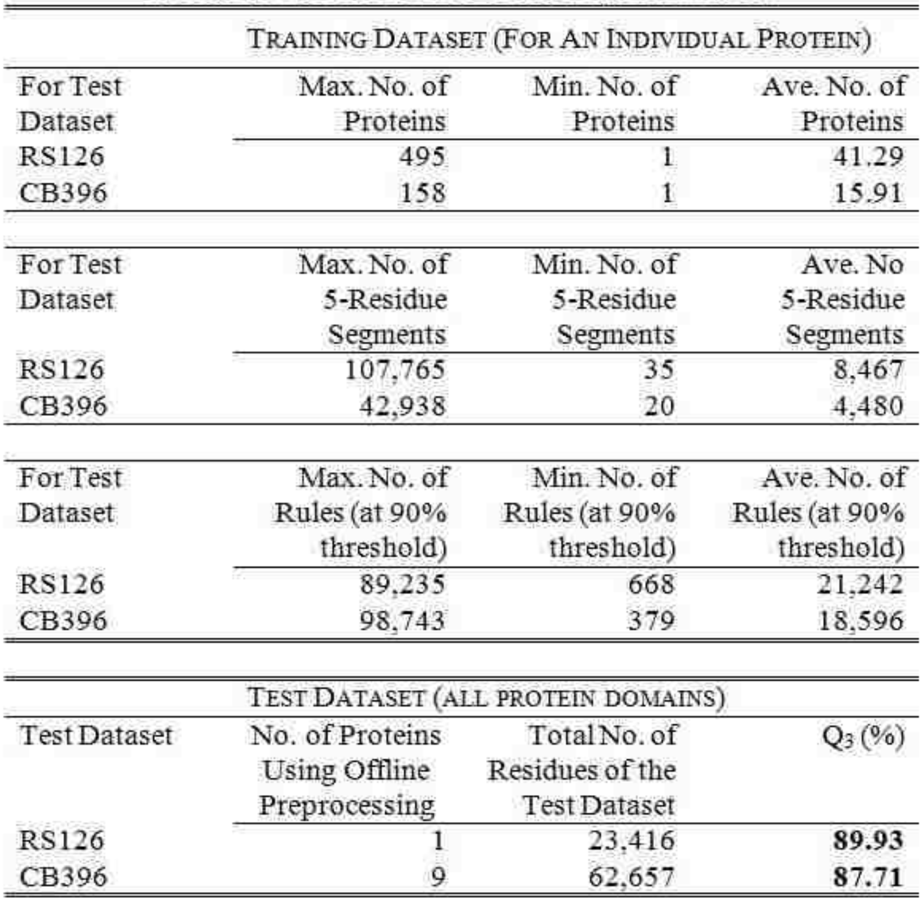
After a BLAST query is made to predict an individual protein, a number of proteins are chosen for data preparation and rule generation as described in Section “Methods.” As shown in Table 6, the maximum number of proteins chosen for a protein prediction from the RS126 and CB396 datasets are 495 and 158, respectively. The minimum number of proteins chosen for a protein prediction from both the RS126 and CB396 datasets are 1. The average number of proteins chosen for a protein prediction from the RS126 set is 41.29, which is larger than the average number of proteins, 15.91, chosen for a protein prediction from the CB396 set.

The proteins chosen are converted to 5-residue segments (five amino acid residues and one secondary structure element) as described in Section “Methods.” As shown in Table 6, the average number of 5-residue segments generated for a protein from the RS126 set is 8,467, which is larger than the average number of 5-residue segments, 4,480, generated for a protein from the CB396 set.

The 5-residue segments are used to generate rules using the RT-RICO algorithm which was discussed in Sections “BLAST-RT-RICO Step 3, Rule Generation” and “Main RT-RICO Rule-Generation Algorithm.” The average number of rules generated for a protein from the RS126 set is only slightly larger than the average number of rules generated for a protein from the CB396 set (21,242 and 18,596, respectively). This is understandable, because the number of rules generated not only depends on the number of 5-residue segment inputs, but also depends on the values of the attributes in the segments.

TABLE 6

PROTEIN SECONDARY STRUCTURE PREDICTION USING BLAST-RT-RICO APPROACH ON RS126 AND CB396 TEST DATASETS



	TRAINING DATASET (FOR AN INDIVIDUAL PROTEIN)		
For Test Dataset	Max. No. of Proteins	Min. No. of Proteins	Ave. No. of Proteins
RS126	495	1	41.29
CB396	158	1	15.91

For Test Dataset	Max. No. of 5-Residue Segments	Min. No. of 5-Residue Segments	Ave. No 5-Residue Segments
RS126	107,765	35	8,467
CB396	42,938	20	4,480

For Test Dataset	Max. No. of Rules (at 90% threshold)	Min. No. of Rules (at 90% threshold)	Ave. No. of Rules (at 90% threshold)
RS126	89,235	668	21,242
CB396	98,743	379	18,596

	TEST DATASET (ALL PROTEIN DOMAINS)		
Test Dataset	No. of Proteins Using Offline Preprocessing	Total No. of Residues of the Test Dataset	Q_3 (%)
RS126	1	23,416	**89.93**
CB396	9	62,657	**87.71**

The number of proteins from the RS126 set using offline processing is 1, and the number of proteins from the CB396 set using offline processing is 9. Thus, in total, only 10 proteins use the rules shown in Table 2.

Cuff and Barton [2] tested the RS126 set with various prediction methods and generated Q_3 scores of 73.5% (PHD), 71.1% (DSC), 70.3% (PREDATOR), 72.7% (NNSSP), and 74.8% for the CONSENSUS method. As shown in Table 6, the BLAST-RT-RICO method has a Q_3 score of 89.93%, which is higher than the Q_3 score generated by other methods using the same RS126 test dataset reported by Cuff and Barton [2].

Cuff and Barton [2] also tested the same prediction methods using the CB396 set, resulting in Q_3 scores of 71.9% (PHD), 68.4% (DSC), 68.6% (PREDATOR), 71.4% (NNSSP) and 72.9% for the CONSENSUS method. As shown in Table 6, the BLAST-

RT-RICO method has a Q_3 score of 87.71%, which is higher than the Q_3 score generated by other methods using the same CB396 test dataset reported in [2].

It is important to note that, because of the different approaches and test design strategies reported in other studies, it is difficult to directly compare prediction results between the BLAST-RT-RICO method presented in this paper and other methods. The final Q_3 scores comparison should be used as a general guide, not a strict percentile comparison.

Conclusions

Presented in this paper was a novel rule-based data mining method, BLAST-RT-RICO, which utilizes data from proteins with significant sequence alignments, and generates rules that can be used in predicting protein secondary structure. The Q_3 scores of 89.93% for the RS126 set and 87.71% for the CB396 set are better than the Q_3 scores that have been reported for comparable computational methods using the same datasets.

The main RT-RICO rule generation algorithm has a time complexity of $O(m^2 2^n)$, with m^2 dominating the time complexity. The current implementation of the algorithm enables the generation of rules from the available protein data within an acceptable timeframe, resulting in efficient prediction of the secondary structure of available test datasets.

Like the artificial neural network methods that have been investigated for predicting protein secondary structure, the BLAST-RT-RICO method makes use of the homologues of proteins and the fundamental principle that structure is more conserved than sequence. Theoretically, when the number of proteins for which the 3D structure has been calculated experimentally increases, the more likely it is that the homologues of proteins can be found, and the more accurate the method may become (with less dependence of offline-processing, which normally produces poorer results).

In the future, we plan to more rigorously examine the training datasets for each test protein. The next natural step would be to construct a BLAST-RT-RICO prediction server with functions to analyze training datasets and prediction results. A server implementation also would make this promising rule-based prediction method more easily accessible to the broader research community.

References

1. Rost B: **Rising accuracy of protein secondary structure prediction.** In *Protein structure determination, analysis, and modeling for drug discovery*. Edited by: Chasman D. New York: Dekker; 2003:207-249.

2. Cuff JA, Barton GJ: **Evaluation and improvement of multiple sequence methods for protein secondary structure prediction.** *Proteins* 1999, **34(4)**: 508–519.

3. Kabsch W, Sander C: **How good are predictions of protein secondary structure?** *FEBS Letters* 1983, **155(2)**: 179-182.

4. Rost B, Sander C: **Prediction of protein secondary structure at better than 70% accuracy.** *J Mol Biol* 1993, **232(2)**: 584-599.

5. King RD, Sternberg MJ: **Identification and application of the concepts important for accurate and reliable protein secondary structure prediction.** *Protein Sci* 1996, **5(11)**: 2298–2310.

6. Frishman D, Argos P: **Seventy-five percent accuracy in protein secondary structure prediction.** *Proteins* 1997, **27(3)**: 329–335.

7. Salamov AA, Solovyev VV: **Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments.** *J Mol Biol* 1995, **247(1):** 11–15.

8. Rost B, Sander C: **Improved prediction of protein secondary structure by use of sequence profiles and neural networks.** *Proc Natl Acad Sci USA* 1993, **90(16)**: 7558–7562.

9. Fadime UY, O¨zlem Y, Metin T: **Prediction of secondary structures of proteinsnext term using a two-stage method.** *Computers & Chemical Engineering* 2008, **32(1-2)**: 78-88.

10. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: **The Protein Data Bank.** *Nucleic Acids Res* 2000, **28(1)**: 235-242.

11. **BLAST: Basic Local Alignment Search Tool** [http://blast.ncbi.nlm.nih.gov/], accessed 30 Nov 2009.

12. Baldi P, Brunak S, Chauvin Y, Andersen CA, Nielsen H: **Assessing the accuracy of prediction algorithms for classification: an overview.** *Bioinformatics* 2000, **16(5)**: 412-424.

13. Zhang CT, Zhang R: **Q9, a content-balancing accuracy index to evaluate algorithms of protein secondary structure prediction.** *Int J Biochem Cell Biol* 2003, **35(8)**: 1256-1262.

14. Jones DT: **Protein secondary structure prediction based on position-specific scoring matrices.** *J Mol Biol* 1999, **292(2)**: 195-202.

15. Bryson K, McGuffin LJ, Marsden RL, Ward JJ, Sodhi JS, Jones DT: **Protein structure prediction servers at University College London.** *Nucleic Acids Res* 2005, **33(Web Server issue)**: W36-38.

16. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.** *Nucleic Acids Res* 1997, **25(17)**:3389-3402.

17. Cuff JA, Barton GJ: **Application of multiple sequence alignment profiles to improve protein secondary structure prediction.** *Proteins* 2000, **40(3)**: 502-511.

18. Hu HJ, Pan Y, Harrison R, Tai PC: **Improved protein secondary structure prediction using support vector machine and a new encoding scheme and an advanced tertiary classifier.** *IEEE Trans NanoBiosci* 2004, **3(4)**: 265–271.

19. Kim H, Park H: **Protein secondary structure prediction based on an improved support vector machines approach.** *Protein Eng* 2003, **16(8)**: 553-560.

20. Nguyen MN, Rajapakse JC: **Two stage support vector machines for protein secondary structure prediction.** *Intl J Data Mining & Bioinformatics* 2007, **1**: 248-269.

21. Levitt M, Chothia C: **Structural patterns in globular proteins.** *Nature* 1976, **261(5561)**: 552-558.

22. Maglia AM, Leopold JL, Ghatti VR: **Identifying Character Non-Independence in Phylogenetic Data Using Data Mining Techniques.** In *Proc Second Asia-Pacific Bioinformatics Conference Dunedin: 2004; New Zealand*.

23. Leopold JL, Maglia AM, Thakur M, Patel B, Ercal F: **Identifying Character Non-Independence in Phylogenetic Data Using Parallelized Rule Induction From Coverings.** In *Data Mining VIII: Data, Text, and Web Mining and Their Business Applications.* WIT Transactions on Information and Communication Technologies; 2007: 45-54.

24. Kabsch W, Sander C: **Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.** *Biopolymers* 1983, 22(12): 2577–2637.

25. Klepeis JL, Floudas CA: **Ab initio prediction of helical segments in polypeptides.** *J Comput Chem* 2002, **23(2)**: 245-266.

26. Han J, Kamber M: *Data Mining: Concepts and Techniques*. Morgan Kaufmann; 2001.

27. Pawlak Z: **Rough Classification.** *Int J Man-Machine Studies* 1984, **20**: 469-483.

28. Grzymala-Busse JW: **Ch.3. Knowledge Acquisition.** In *Managing Uncertanity in Expert System*. Boston: Kluwer Academic; 1991: 43-76.

PAPER

5. RULE VISUALIZATION OF PROTEIN MOTIF SEQUENCE DATA FOR SECONDARY STRUCTURE PREDICTION

Leong Lee[1], Jennifer L. Leopold[1], Patrick G. Edgett[1], Ronald L. Frank[2]

[1]Department of Computer Science, [2]Department of Biological Sciences
Missouri University of Science and Technology, Rolla, Missouri, USA

Abstract

Protein secondary structure prediction has been a well studied research problem in bioinformatics for years. In previous papers, we presented a rule-based data mining method called RT-RICO (Relaxed Threshold Rule Induction from Coverings) that addressed this problem. Our method surpassed the accuracy, or Q_3 score, that had been reported for other computational methods for protein secondary structure prediction using the standard datasets, RS126 and CB396. The success of our rule-based method supported the belief that there are meaningful statistical relationships between any secondary structure position and its neighboring amino acids. However, because of the vast amount of rules generated by RT-RICO, potentially useful information within a rule set was difficult to identify. Herein we discuss the results of examining those RT-RICO rules using an existing association rule visualization tool, modified to account for the non-Boolean characterization of protein secondary structure.

1. Introduction

Prediction of the 3D structure of a protein from its amino acid sequence is a very challenging research goal in bioinformatics, and has been studied extensively since the

1960s. Rost (2003) suggests that protein 3D structure prediction from sequence cannot be achieved fully. However, research has continuously improved computational methods for predicting simplified aspects of structure.

It is not an easy task to evaluate the performance of a protein secondary structure prediction method. In particular, the use of different datasets for training and testing each algorithm makes it difficult to find an objective comparison of methods (Cuff and Barton, 1999). Rost (2003) stated that “there is no value in comparing methods evaluated on different datasets.” Therefore, efforts have been made to develop standard test datasets to accurately evaluate the performance of different prediction methods. Rost and Sander (1993) selected a list of 126 protein domains that now constitutes one comparative standard (the RS126 dataset). Cuff and Barton (1999) described the development of a non-redundant test set of 396 protein domains (the CB396 dataset), where no two proteins in the set share more than 25% sequence identity over a length of more than 80 residues (Rost and Sander, 1993). They used the CB396 set to test four secondary structure prediction methods: PHD (Rost and Sander, 1993), DSC (King and Sternberg, 1996), PREDATOR (Frishman and Argos, 1997) and NNSSP (Salamov and Solovyev, 1995). They also combined the four methods by a simple majority-wins method, the CONSENSUS method (Cuff and Barton, 1999). The resulting accuracy, or Q_3 scores, for the CB396 set were 71.9% (PHD), 68.4% (DSC), 68.6% (PREDATOR), 71.4% (NNSSP), and 72.9% for the CONSENSUS method. In the same research study, Cuff and Barton (1999) also tested the RS126 set, in which the Q_3 scores were 73.5% (PHD), 71.1% (DSC), 70.3% (PREDATOR), 72.7% (NNSSP), and 74.8% for the CONSENSUS method.

An interesting secondary structure prediction method is described by Fadime et al. (2008), wherein a two-stage approach is taken to address the problem. In the first stage, the folding type of a protein is determined (i.e., “all-α”, “all-β”, “α/β”, or “α+β”). The second stage utilizes data from the Protein Data Bank (PDB) (Berman et al., 2000) and a probabilistic search algorithm to determine the locations of secondary structure elements. The resulting average accuracy of their prediction score is 74.1%. This two-stage method indicated that there are statistical relationships between a secondary structure element and its neighboring amino acid residues.

Protein secondary structure is defined by specific clusters of hydrogen bonds between the C=O and N-H of the backbone peptide bond within a polypeptide chain. Although certain amino acids are associated with secondary structure more often than others, no simple rule exists to predict whether or not a short string of amino acids will form the appropriate structure. However, since secondary structure is a local organization in the peptide chain, the likelihood that a particular amino acid is part of a helix or beta structure is dependent upon its neighboring amino acids. Generating rules from many examples of known secondary structure can provide a more accurate prediction of the structural tendencies in a particular segment of the chain.

In (Lee et al., 2010a) we introduced a rule-based prediction approach, RT-RICO (Relaxed Threshold Rule Induction from Coverings), that takes advantage of the fact that different protein folding types have different chemical structures; hence the statistical relationships between a secondary structure element and its neighboring amino acid residues also should be different among these classes. RT-RICO discovers these relationships by generating rules that can be used to predict secondary structure. The resulting Q_3 score was 81.75% on the RS126 set, and 79.19% on the CB396 set.

In (Lee et al., 2010b) we presented a slightly modified method for predicting the secondary structure elements called BLAST-RT-RICO (Relaxed Threshold Rule Induction from Coverings). First, a query using the Web-based NCBI/PSI-BLAST search engine is performed for a protein (BLAST, 2009). Suitable proteins with significant multiple sequence alignments are identified. Then the RT-RICO algorithm is used to generate rules representing dependencies between protein amino acid sequences and the related secondary structure elements. The BLAST-RT-RICO method performed better than our previously developed method, with a Q_3 accuracy of 89.93% on the RS126 set and 87.71% on the CB396 set.

For these research studies thousands of rules were generated. Despite the large volume of output, it was noticeable that different protein type classes generated different type of rules. It was also logical (based on successful test results) to conclude that for each test protein query, the NCBI/PSI-BLAST search engine returned sets of proteins that produced different sets of rules. Yet, because of the vast amount of rules, it was not only infeasible to visualize them, but also impractical to compare different sets of rules.

Wong et al. (1999) presented a technique to visualize association rules. Their procedure can handle hundreds of multiple antecedent association rules in a 3D display with minimum human interactions. However, this tool was designed to handle only Boolean-valued association rules (Han and Kamber, 2001) (i.e., rules concerning only the presence or absence of attributes). The rules generated from (Lee et al., 2010a) and (Lee et al., 2010b) are multi-valued. Therefore, we slightly modified the Wong technique in order to visualize and compare the rule sets generated from different protein type classes that were determined in (Lee et al., 2010a and Lee et al., 2010b). As will be discussed in Section 3, the rule visualization tool facilitated analysis of the rule sets from different perspectives, and led to consideration of new relationships between protein secondary structure elements and their neighboring amino acids.

2. Related Work

To better understand the challenges of rule visualization of protein motif sequence data, we first need to explain how the rules are generated, and how they are used to address the protein secondary structure prediction problem.

2.1. Protein Secondary Structure Prediction Problem Description

Protein secondary structure prediction requires that a data sequence D be compared to a prediction result sequence M to calculate the Q_3 (prediction accuracy) score (Baldi et al., 2000); that is:

- Input: Amino acid sequence, $A = a_1, a_2, \ldots a_N$; Data for comparison, $D = d_1, d_2, \ldots d_N$

  a_i is an element of a set of 20 amino acids, $\{A,R,N\ldots V\}$

  d_i is an element of a set of secondary structures, $\{H,E,C\}$, which represents helix H, sheet E, and coil C.

- Output: Prediction result: $M = m_1, m_2, \ldots m_N$

  m_i is an element of a set of secondary structures, $\{H,E,C\}$

- Q_3 Score (Cuff and Barton, 1999), to assess the accuracy of the predictions:

  $Q_3 = \sum_{(i=H,E,C)} predicted_i \,/\, observed_i \times 100$

2.2. Other Prediction Methods

Rost (2003) classifies protein secondary structure prediction methods into three generations. The first generation methods depend on single residue statistics to perform prediction. The second generation methods depend on segment statistics. The third generation methods use evolutionary information to predict secondary structure.

Many of the third generation methods exploit our knowledge about multiple sequence alignments through neural network designs, or more recently, support vector machine designs. This has resulted in a significant increase in prediction accuracy (to nearly 80%). One of the primary assumptions that these techniques use is that the full distribution of amino acids occurs at a particular (secondary structure) position and its vicinity; typically there are approximately seven amino acid residues on either side due to evolution. This evolution-based knowledge is obtained by searching existing protein databases using multiple sequence alignment algorithms. From the success of these prediction methods we can deduce that there are relationships between any secondary structure element (at a particular position) and its neighboring amino acids. Although this neighboring vicinity definition differs somewhat among various methods, the general relationships are captured by trained neural networks, resulting in the high accuracy of some third generation methods.

Levitt and Chothia (1976) proposed to classify proteins as four basic types or classes according to their α-helix and β-sheet content. "All-α" class proteins consist almost entirely (at least 90%) of α-helices. "All-β" class proteins are composed mostly of β-sheets (at least 90%). The "α/β" class proteins have alternating, mainly parallel segments of α-helices and β-sheets. The "α+β" class proteins have a mixture of all-α and all-β regions, mostly in sequential order. The first stage of the two-stage method developed by Fadime et al. (2008) is able to determine the class of unknown proteins with 100% accuracy. In the second stage they use a probabilistic approach based on their stage one results. The amino acid sequences of the training set are distributed into overlapping sequence groups of three to seven residues. These groups then are used to calculate the probability statistics for secondary structure. Specifically, the secondary structure at a particular sequence location is determined by comparing the probabilities that an amino acid residue is a particular secondary structure type based on the statistics.

This greatly simplifies part of the protein secondary structure prediction problem; if it can be determined which one of the four classes a protein belongs to, other approaches can be applied to predict the secondary structure elements within the four classes. Hence, for each protein type class, there are statistical relationships between a secondary structure element and its neighboring amino acid residues.

2.3. Rule-Based RT-RICO

We developed a rule-based secondary structure prediction method called RT-RICO. The detailed algorithms are given in (Lee et al., 2010a); here we simply provide an overview of how we derive and use the generated rules.

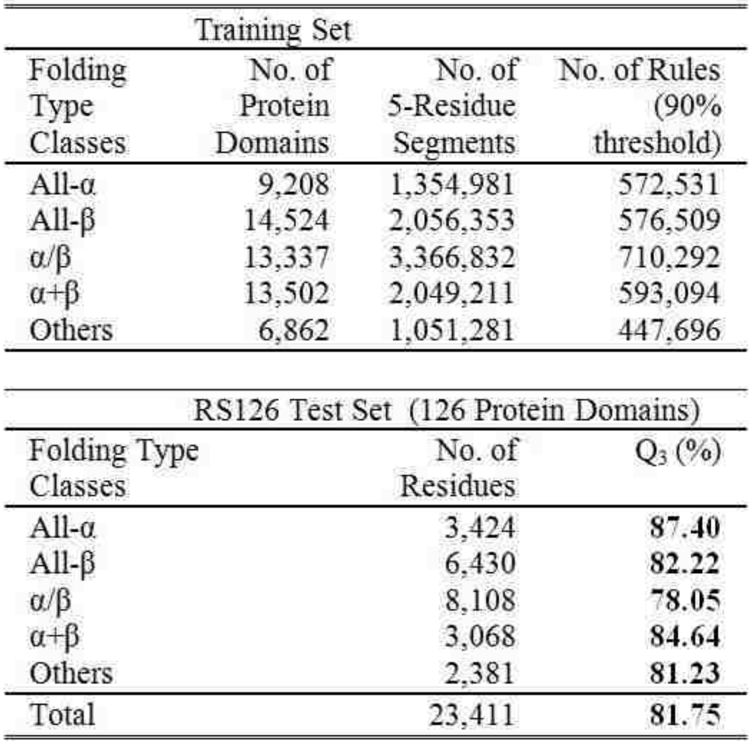
2.3.1. RT-RICO Step 1

At step 1, data preparation, all protein names and corresponding folding types of each protein are retrieved from the SCOP database (Andreeva et al., 2008) (Murzin et al. 1995). All available corresponding protein sequences and secondary structure sequences are obtained from the PDB database (Berman et al., 2000). Five databases of protein domains (with their amino acid sequences and secondary structure sequences) of different protein domain types (e.g., “all-α”, “all-β”, “α/β”, “α+β” and “others”) are built. Proteins from the test datasets (RS126 or CB396) are first removed from these databases, so that they will be excluded from the possible training datasets. Protein domains from different protein families are selected to form the training datasets. See Table I for the number of protein domains in each training dataset derived from the RS126 test dataset.

The protein secondary structure sequences from PDB are formed by elements of eight states of secondary structure, {H, G, I, E, B, T, S, -}. The eight states are converted to four states to facilitate rule generation as follows: (G, H, I) => Helix H; (E, B) => Sheet E; (T, S) => Coil C; (-) => “-.” Note that rule generation uses a four-state decision attribute. The final Q_3 score calculation uses a three-state decision attribute: (G, H, I) => Helix H; (E, B) => Sheet E; (Rest) => Coil C.

Klepeis and Floudas (2002) showed that the use of overlapping segments of five residues is very effective in predicting the helical segments of proteins. Thus, the overlapping 5-residue segments approach was used to prepare the RT-RICO training data records. As shown in Fig. 1, for each secondary structure element, five neighboring amino acid residues are extracted to form a segment of five amino acid residues, plus one secondary structure element. These segments are used as input to the RT-RICO rule generation algorithm (discussed in detail in (Lee et al., 2010a)). The numbers of 5-residue segments generated for the five protein type classes are shown in Table I.

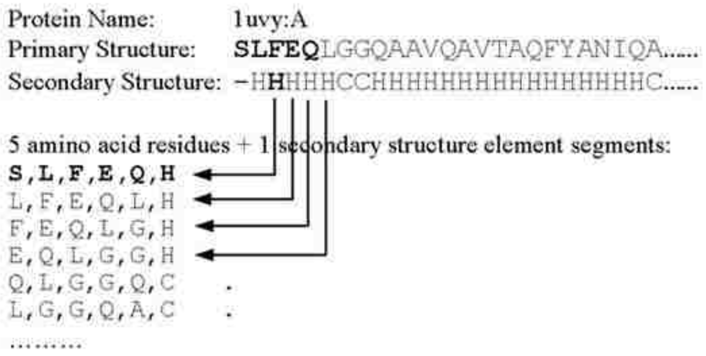
TABLE I
PROTEIN SECONDARY STRUCTURE PREDICTION USING RT-RICO RULE GENERATION ON RS126 TEST DATASET



Training Set			
Folding Type Classes	No. of Protein Domains	No. of 5-Residue Segments	No. of Rules (90% threshold)
All-α	9,208	1,354,981	572,531
All-β	14,524	2,056,353	576,509
α/β	13,337	3,366,832	710,292
α+β	13,502	2,049,211	593,094
Others	6,862	1,051,281	447,696

RS126 Test Set (126 Protein Domains)		
Folding Type Classes	No. of Residues	Q_3 (%)
All-α	3,424	**87.40**
All-β	6,430	**82.22**
α/β	8,108	**78.05**
α+β	3,068	**84.64**
Others	2,381	**81.23**
Total	23,411	**81.75**

The main inputs to the RT-RICO rule generation algorithm are in the form of 6-tuples. The first five elements of a 6-tuple are formed by amino acid residues, {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. The last element of a 6-tuple is formed by one of four secondary structure states {H, E, C, -}. The last element is considered the decision attribute. In other words, the input to step 2 of RT-RICO, rule generation, is in the form of an $m \times (n+1)$ matrix, where m is the number of all entities (the number of 5-residue plus one secondary structure element segments), and $n = 5$ in this case.



Protein Name: 1uvy:A

Primary Structure: **SLFEQ**LGGQAAVQAVTAQFYANIQA......

Secondary Structure: –H**H**HHHCCHHHHHHHHHHHHHHHHC......

5 amino acid residues + 1 secondary structure element segments:

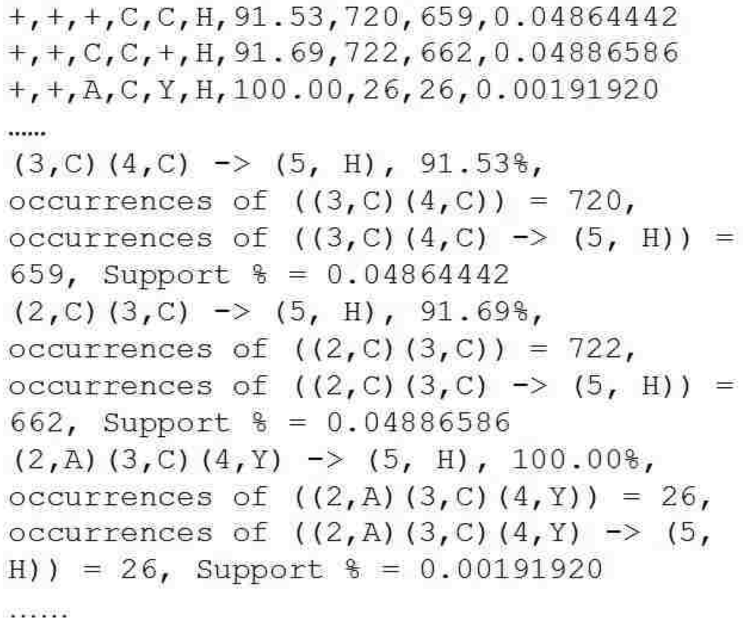
S,L,F,E,Q,H
L,F,E,Q,L,H
F,E,Q,L,G,H
E,Q,L,G,G,H
Q,L,G,G,Q,C
L,G,G,Q,A,C
....

Fig. 1. Protein primary structure 5-residue segments and related secondary structure elements representation. Note: The first and second positions at the beginning of the sequence are represented by 3 residues + 1, and 4 residues + 1 segments, respectively. They form separate training datasets.

2.3.2. RT-RICO Step 2

RT-RICO generates rules based on the segments in the form of an *m×(n+1)* matrix. Some examples of the generated rules are shown in Fig. 2 in two separate formats. The first format is intended to be read by the computer programs at the later prediction stage (i.e., the computer rule format). The second format is intended to be read by the user (i.e., the human rule format). The first rule is interpreted as follows: if the fourth position attribute is “C”, and the fifth position attribute is “C”, then the sixth (decision) attribute is “H” with a confidence of 91.53% and a support of 0.04864442% (where the support is calculated from the “hits” shown, 659 / number of all inputs (5-residue segments)). Confidence and support are defined in (Han and Kamber, 2001).

The corresponding first rule is interpreted as follows: if the first position attribute is “+” (representing any amino acid element), the second position attribute is “+”, the third position attribute is “+”, the fourth position attribute is “C”, and the fifth position attribute is “C”, then the sixth attribute (i.e., the decision attribute) is “H.” The number of occurrences of the fourth position attribute (which is “C”) and the fifth position attribute (which is “C”) equals 720 among all inputs to RT-RICO. The number of occurrences of the fourth position attribute (which is “C”), the fifth position attribute (which is “C”), and the sixth attribute (which is “H”), equals 659 among all inputs to RT-RICO. The confidence is 91.53% and the support is 0.04864442%.



```
+,+,+,C,C,H,91.53,720,659,0.04864442
+,+,C,C,+,H,91.69,722,662,0.04886586
+,+,A,C,Y,H,100.00,26,26,0.00191920
......
(3,C)(4,C) -> (5, H), 91.53%,
occurrences of ((3,C)(4,C)) = 720,
occurrences of ((3,C)(4,C) -> (5, H)) =
659, Support % = 0.04864442
(2,C)(3,C) -> (5, H), 91.69%,
occurrences of ((2,C)(3,C)) = 722,
occurrences of ((2,C)(3,C) -> (5, H)) =
662, Support % = 0.04886586
(2,A)(3,C)(4,Y) -> (5, H), 100.00%,
occurrences of ((2,A)(3,C)(4,Y)) = 26,
occurrences of ((2,A)(3,C)(4,Y) -> (5,
H)) = 26, Support % = 0.00191920
......
```

Fig. 2. Sample rules generated by RT-RICO.

2.3.3. RT-RICO Step 3

In its final step, RT-RICO loads protein primary structures from the test dataset, and predicts the secondary structure elements. As shown in Fig. 3, for each secondary structure element prediction position, five neighboring amino acid residues are extracted to form a segment of residues. Each of these segments is compared with the generated rules (generated from 5-residue segments). If a segment matches a rule, the support value of the rule is taken into consideration for the prediction of the related secondary structure element. The algorithm first searches for matching rules with 100% confidence value. The secondary structure element with the highest total support value (among 100% confidence value rules) is selected. If no matching rule exists among 100% confidence value rules, the algorithm then searches for other matching rules (with confidence values greater than or equal to 90%, but less than 100%). The secondary structure element with the highest total support value among these rules is selected as the predicted secondary structure element for that specific position. If no matching rule is found for the segment, the secondary structure of the previous position is used as the predicted structure.

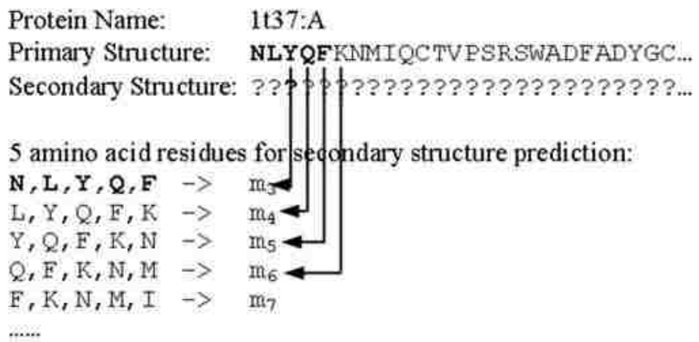

Protein Name: 1t37:A
Primary Structure: NLYQFKNMIQCTVPSRSWADFADYGC...
Secondary Structure: ??????????????????????????...
5 amino acid residues for secondary structure prediction:
N,L,Y,Q,F -> m_3
L,Y,Q,F,K -> m_4
Y,Q,F,K,N -> m_5
Q,F,K,N,M -> m_6
F,K,N,M,I -> m_7
......

Fig. 3. Protein primary structure 5-residue segments and related secondary structure elements prediction. m_i is an element of set {H,E,C,-}. It is then converted to an element of the set {H, E, C}. Note: The first and second positions at the beginning of the sequence are represented by 3 residue, and 4 residue segments.

Table I lists the number of protein domains in each training dataset and the performance of the RT-RICO prediction method on the RS126 test dataset. Table II shows the number of protein domains in each training dataset and the performance of the RT-RICO on the CB396 test dataset. The Q_3 scores are 81.75% for the RS126 set and 79.19% for the CB396 set. Note that a large number of rules are generated; for example, 570,580 rules are generated for the all-α class of the CB396 set (Table II).

In addition to knowing the Q_3 score, we thought it would be interesting to compare the rules from different classes (e.g., all-α class rules compared to the all-β class rules); different classes should produce different rule sets. However, this required an effective method to visualize and compare the numerous RT-RICO rules.

2.4. BLAST-RT-RICO

After the development of RT-RICO, we developed an improved secondary structure prediction method, BLAST-RT-RICO; the detailed algorithms are presented in (Lee et al., 2010b).

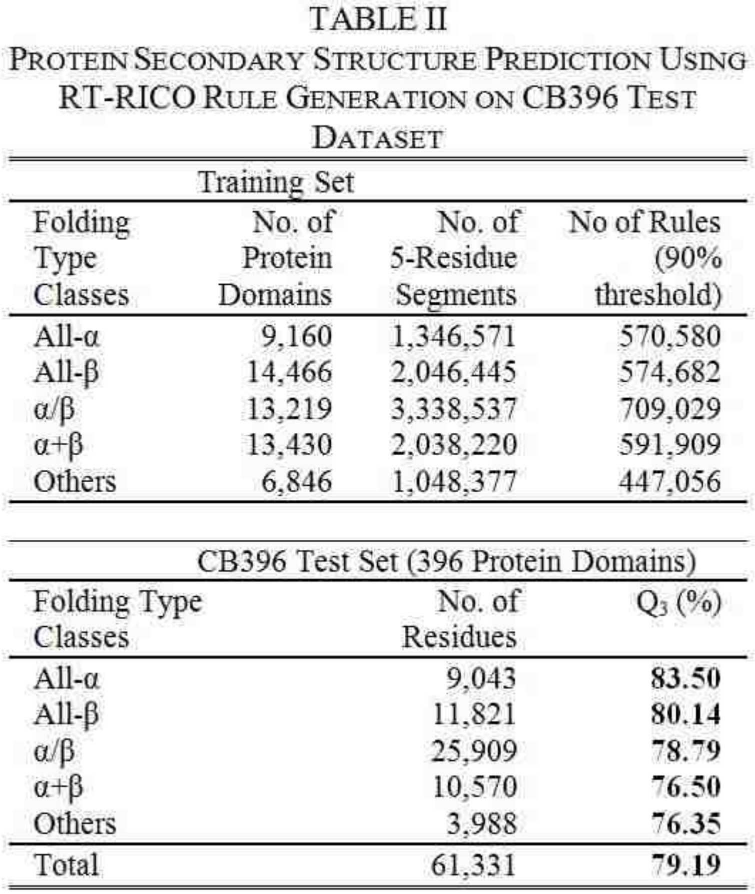


TABLE II
PROTEIN SECONDARY STRUCTURE PREDICTION USING RT-RICO RULE GENERATION ON CB396 TEST DATASET

	Training Set		
Folding Type Classes	No. of Protein Domains	No. of 5-Residue Segments	No of Rules (90% threshold)
All-α	9,160	1,346,571	570,580
All-β	14,466	2,046,445	574,682
α/β	13,219	3,338,537	709,029
α+β	13,430	2,038,220	591,909
Others	6,846	1,048,377	447,056

	CB396 Test Set (396 Protein Domains)	
Folding Type Classes	No. of Residues	Q_3 (%)
All-α	9,043	**83.50**
All-β	11,821	**80.14**
α/β	25,909	**78.79**
α+β	10,570	**76.50**
Others	3,988	**76.35**
Total	61,331	**79.19**

2.4.1. BLAST-RT-RICO Step 1

At step 1, online BLAST & PDB data match, given an input, test protein *A*, *A* = a_1, a_2, … a_N, a BLAST search is performed using *A* as the query sequence. The BLAST search returns a list of proteins with significant sequence alignments and corresponding BLAST scores. Proteins with a score less than or equal to 30 are removed from the list. The test protein *A* is also removed if it appears in the list, so that it will be excluded from the training dataset. A query is first performed to check if any protein from the list already has a known secondary structure record from the PDB database. If this is the case, then the proteins with corresponding secondary structure records are retrieved; they form the inputs to the next step, data preparation.

If a protein from the list does not have a known secondary structure record in the PDB database, the prediction for that protein needs to be handled slightly differently; namely, it will require data from offline preprocessing. These operations can be performed offline because it is not necessary to perform rule generation for every protein prediction. Instead, rules can be generated once and used for all the proteins falling into this category. In offline preprocessing, all proteins and corresponding secondary structure sequences from the PDB database are downloaded to form an initial dataset. Proteins from the test datasets (RS126 or CB396) are first removed, so that they will be excluded from the possible training datasets. Protein domains from different protein families are selected to form the training datasets. See Table III for the number of protein domains in each training dataset for the RS126 and CB396 test datasets. The reason for having two different training datasets is because the RS126 and CB396 data first need to be removed from the initial dataset.

Again, the number of rules generated is considerable; for example, 955,625 rules are generated for the RS126 set. The large size of the rule set is due to the fact that almost all proteins with known secondary structures are used.

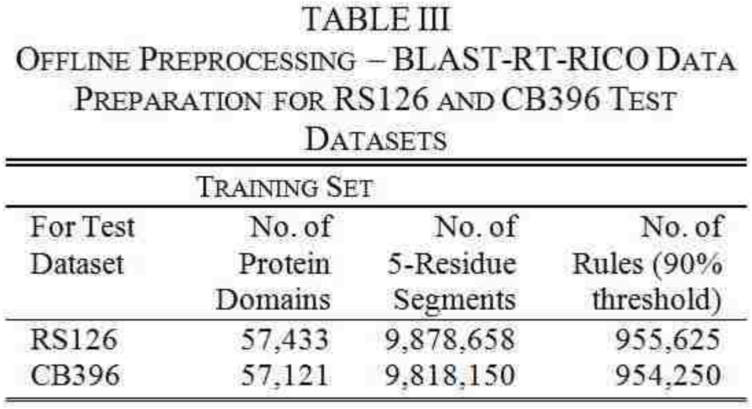


TABLE III
OFFLINE PREPROCESSING – BLAST-RT-RICO DATA PREPARATION FOR RS126 AND CB396 TEST DATASETS

	TRAINING SET		
For Test Dataset	No. of Protein Domains	No. of 5-Residue Segments	No. of Rules (90% threshold)
RS126	57,433	9,878,658	955,625
CB396	57,121	9,818,150	954,250

2.4.2. BLAST-RT-RICO Step 2

The proteins with significant sequence alignments and corresponding secondary structure records are inputs to the data preparation step. For test protein *A*, there is a set of protein primary structure sequence B_i and a set of corresponding secondary structure sequence C_i where $B_i \in \{B_1, B_2, B_3, B_4, \ldots B_p\}$, $C_i \in \{C_1, C_2, C_3, C_4, \ldots C_p\}$, the protein primary structure sequence is $B_i = b_{i,1}, b_{i,2}, b_{i,3}, \ldots b_{i,q_i}$ and the corresponding secondary structure sequence is $C_i = c_{i,1}, c_{i,2}, c_{i,3}, \ldots c_{i,q_i}$ Sequences B_1 to B_p are not necessarily of the same length because they represent different proteins; in other words, sequence *i* has length q_i. Here each $b_{i,j}$ is an element of a set of 20 amino acids*, {A, R, N, … V}*. Initially,

$c_{i,j}$ is an element of a set of eight-state secondary structures, *{H, G, I, E, B, T, S, -}*, as represented in the PDB database. It is then converted to an element of a set of four-state secondary structures, *{H, E, C, -}*.

The same overlapping 5-residue segments approach is used to prepare the training data records. As shown in Fig. 1, for each secondary structure element, five neighboring amino acid residues are extracted to form a segment of amino acid residues, plus one secondary structure element. These segments are used as input to the step 3, rule generation. If B_i is the primary structure sequence, C_i is the secondary structure sequence shown in Fig. 1, and the length of the sequence(s) is q_i, then each 5-residue segment is of the form: $b_{i,j-2}$, $b_{i,j-1}$, $b_{i,j}$, $b_{i,j+1}$, $b_{i,j+2}$, $c_{i,j}$; and j has a value from 3 to $(q_i - 2)$. This data preparation step is performed for all B_i and C_i pairs, where i is from 1 to p.

2.4.3. BLAST-RT-RICO Step 3

RT-RICO generates rules based on the segments in the form of an $m\times(n+1)$ matrix. Some examples of the generated rules are shown in Fig. 2. In BLAST-RT-RICO, for each test protein *A*, a different set of rules is generated, and this set of rules is only used for the prediction of test protein *A*.

2.4.4. BLAST-RT-RICO Step 4

BLAST-RT-RICO loads protein primary structures from the test dataset (a single protein *A* for this case), and predicts the secondary structure elements. As shown in Fig. 3, for each secondary structure element prediction position (for a corresponding amino acid sequence of length *k*, from position *3* to *k-2*), five neighboring amino acid residues are extracted to form a segment of five residues. Each of these segments is compared with the generated rules (generated from 5-residue segments). The rule matching algorithm is the same as the algorithm described in Section 2.3.3 for step 3 of RT-RICO.

Note that as mentioned in Section 2.3.3, the primary (main) selection/sorting criteria is the "confidence" of rules ("support" is the secondary selection criteria). By using "confidence" as the main selection/sorting criteria (instead of using "support"), we eliminate potential errors caused by a misleadingly large support value due to data availability (e.g. orthologs).

Table III lists the number of protein domains, segments, and rules in the training datasets for offline preprocessing. Table IV shows a summary of the number of proteins, segments, and rules in each training dataset (the results of BLAST and subsequent operations) for individual proteins; it also shows the performance of the BLAST-RT-RICO method on the RS126 and CB396 test datasets. The Q_3 scores are 89.93% for the RS126 set and 89.71% for the CB396 set. The average number of rules generated for a protein from the RS126 set is only slightly larger than the average number of rules generated for a protein from the CB396 set, 21,242 and 18,596, respectively.

As with the RT-RICO results, we thought it would be interesting to compare the rules for different test proteins (e.g. RS126 and CB396 sets produces hundreds of rule sets). Clearly, different test proteins should produce different rule sets.

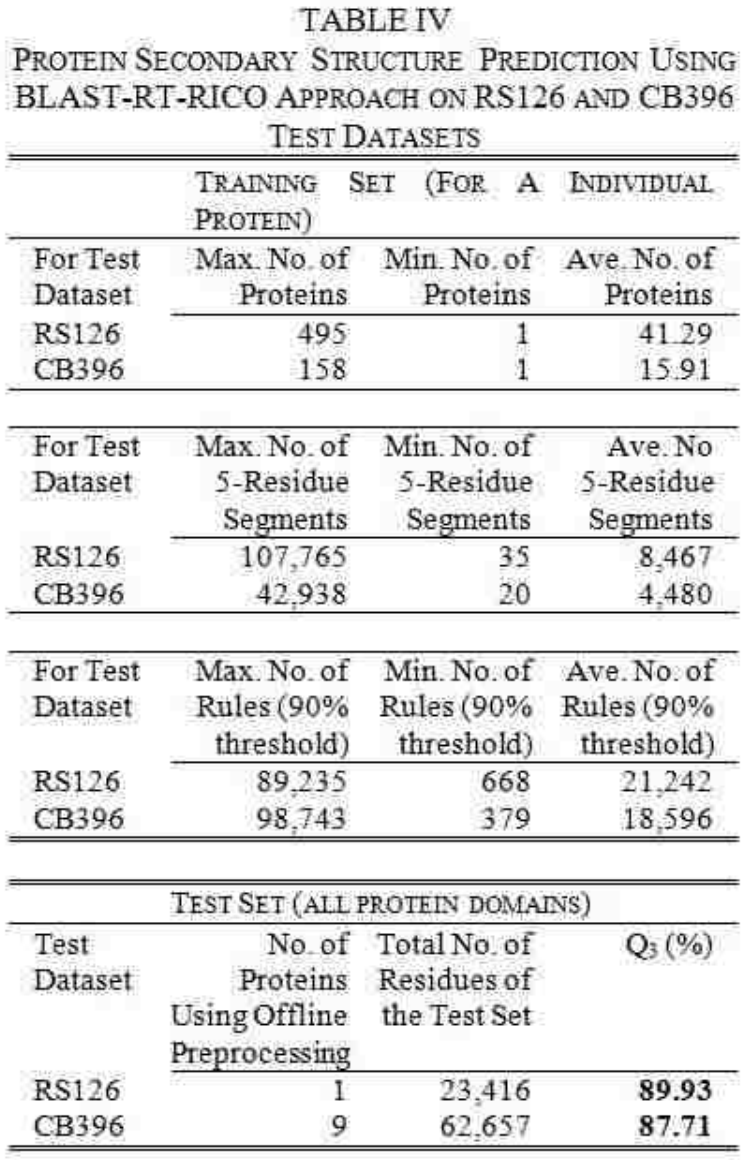


TABLE IV
PROTEIN SECONDARY STRUCTURE PREDICTION USING BLAST-RT-RICO APPROACH ON RS126 AND CB396 TEST DATASETS

	TRAINING SET (FOR A INDIVIDUAL PROTEIN)		
For Test Dataset	Max. No. of Proteins	Min. No. of Proteins	Ave. No. of Proteins
RS126	495	1	41.29
CB396	158	1	15.91

For Test Dataset	Max. No. of 5-Residue Segments	Min. No. of 5-Residue Segments	Ave. No 5-Residue Segments
RS126	107,765	35	8,467
CB396	42,938	20	4,480

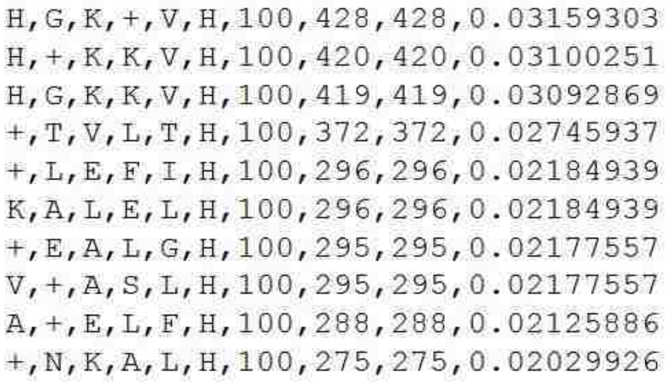
For Test Dataset	Max. No. of Rules (90% threshold)	Min. No. of Rules (90% threshold)	Ave. No. of Rules (90% threshold)
RS126	89,235	668	21,242
CB396	98,743	379	18,596

	TEST SET (ALL PROTEIN DOMAINS)		
Test Dataset	No. of Proteins Using Offline Preprocessing	Total No. of Residues of the Test Set	Q_3 (%)
RS126	1	23,416	**89.93**
CB396	9	62,657	**87.71**

3. Rule Visualization

An association rule in data mining is an implication of the form X → Y where X is a set of antecedent items, and Y is the consequent item (Wong et al., 1999). Wong et al. (1999) developed a technique to visualize hundreds of multiple antecedent association rules in a three-dimensional display. However, Wong's technique was designed to handle only Boolean association rules (Han and Kamber, 2001), rules concerning only the presence or absence of attributes. The association rules generated from (Lee et al., 2010a) and (Lee et al., 2010b) for protein secondary structure are multi-valued, and hence considered quantitative (Han and Kamber, 2001).

We see in Table I that there are 572,531 rules generated by the "all-α" class training set. These rules are sorted by confidence value, then by support value. They are sorted this way because during the prediction steps of RT-RICO and BLAST-RT-RICO, the algorithms first search for matching rules with 100% confidence value. Then the secondary structure element with the highest total support value is selected. So the top 10 rules have 100% confidence value, and the highest support values (see Fig. 4).



```
H,G,K,+,V,H,100,428,428,0.03159303
H,+,K,K,V,H,100,420,420,0.03100251
H,G,K,K,V,H,100,419,419,0.03092869
+,T,V,L,T,H,100,372,372,0.02745937
+,L,E,F,I,H,100,296,296,0.02184939
K,A,L,E,L,H,100,296,296,0.02184939
+,E,A,L,G,H,100,295,295,0.02177557
V,+,A,S,L,H,100,295,295,0.02177557
A,+,E,L,F,H,100,288,288,0.02125886
+,N,K,A,L,H,100,275,275,0.02029926
```

Fig. 4. Top 10 association rules generated by "all-α" class training set for the RS126 set.

The first rule can be interpreted as (0,H) (1,G) (2,K) (4,V) → (5, H), with 100% confidence, and 0.03159303% support. It is considered a quantitative rule because it states that if position 0 is amino acid H, position 1 is amino acid G, position 2 is amino acid K, and position 4 is amino acid V, then the decision attribute (i.e., secondary structure element) is H (Helix). These rules can be visualized by using a modified version of Wong's technique. Instead of using different colors to distinguish between the antecedent and consequent items, we use different colors to represent different amino acids and different secondary structure elements. Because positions 0 to 4 always represent amino acid residues, and position 5 is the decision attribute representing the secondary structure element, there is no need to distinguish between the different types of items; positions 0 to 4 are antecedent items and position 5 is the only consequent item for our application.

A visualization of the top 30 association rules generated by the "all-α" class training set for the RS126 sets is shown in Fig. 5. Note that the top 10 rules from Fig. 4 can be found on the left side of the 3D diagram in Fig. 5. The confidence values are not shown because they are always 100% for the top 30 rules. A few interesting facts become obvious upon examining the 3D diagram. First, only 15 different amino acids (instead of 20) appear in the top 30 rules. Secondly, all decision attribute values at position 5 are "H/Helix." This may not be surprising, because the rules are generated from the "all-α" class. But the 3D diagram makes visualization of these facts much easier to observe. We also become motivated to compare color patterns between different rule sets, which will be discussed in the next section.

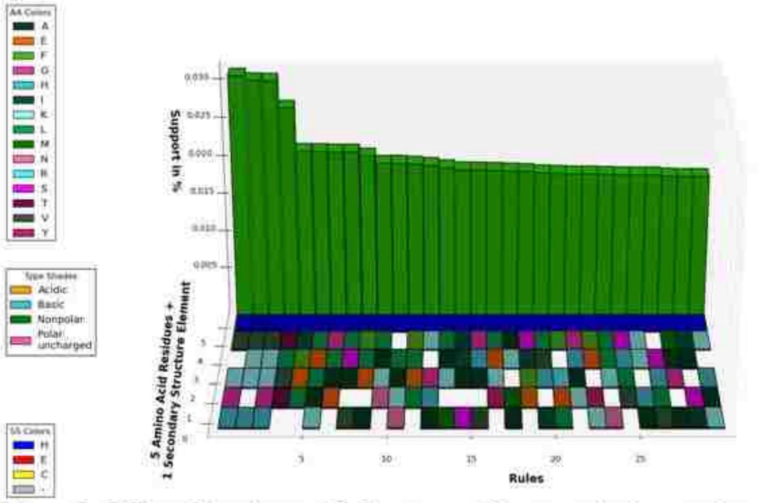


Fig. 5. Visualization of the top 30 association rules generated by "all-α" class training set for the RS126 set (color by type).

One significant advantage of using this technique to analyze amino-acid attributes and a secondary structure decision attribute is that we can change the amino acids' colors (or any attribute's color) in the 3D diagrams to represent different properties. In Fig. 5 the amino acid colors were chosen according to the different amino acid types (e.g., acidic, basic, nonpolar, and polar uncharged). As shown in Table V, amino acids belonging to the same type use similar color shades (acidic: orange; basic: teal; nonpolar: green; polar uncharged: pink). This is very useful when we want to examine certain chemical properties. For example, colors can be changed to distinguish amino acids of different sizes (e.g., Fig. 10 and Fig. 11), or other relevant chemical properties. The rules in the (color by type) 3D diagrams are sorted by secondary structure elements, decision attribute position 5 (in the order of H, E, C, -), and then support.

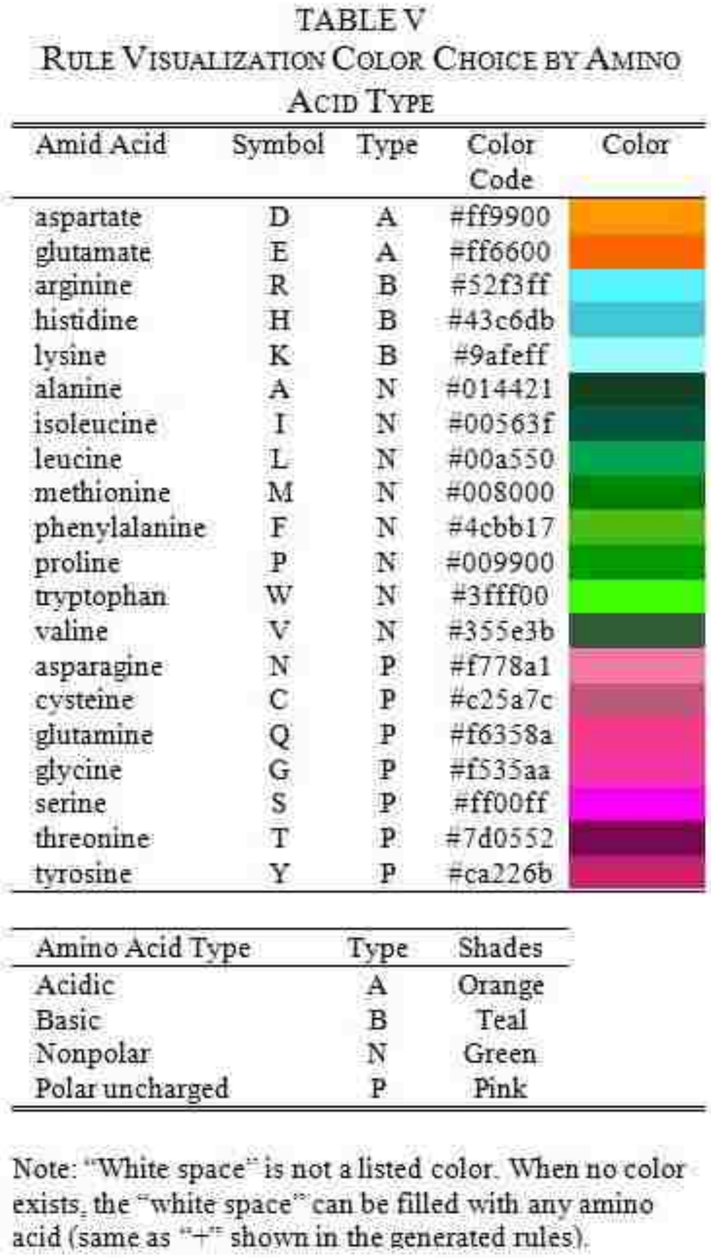


TABLE V
RULE VISUALIZATION COLOR CHOICE BY AMINO ACID TYPE

Amid Acid	Symbol	Type	Color Code	Color
aspartate	D	A	#ff9900	🟧
glutamate	E	A	#ff6600	🟧
arginine	R	B	#52f3ff	🟦
histidine	H	B	#43c6db	🟦
lysine	K	B	#9afeff	🟦
alanine	A	N	#014421	🟩
isoleucine	I	N	#00563f	🟩
leucine	L	N	#00a550	🟩
methionine	M	N	#008000	🟩
phenylalanine	F	N	#4cbb17	🟩
proline	P	N	#009900	🟩
tryptophan	W	N	#3fff00	🟩
valine	V	N	#355e3b	🟩
asparagine	N	P	#f778a1	🟪
cysteine	C	P	#c25a7c	🟪
glutamine	Q	P	#f6358a	🟪
glycine	G	P	#f535aa	🟪
serine	S	P	#ff00ff	🟪
threonine	T	P	#7d0552	🟪
tyrosine	Y	P	#ca226b	🟪

Amino Acid Type	Type	Shades
Acidic	A	Orange
Basic	B	Teal
Nonpolar	N	Green
Polar uncharged	P	Pink

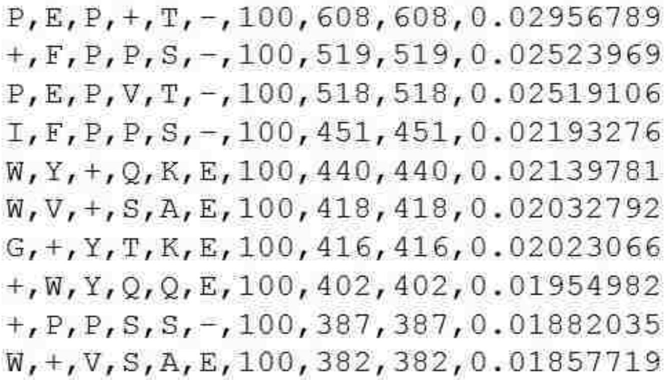
Note: "White space" is not a listed color. When no color exists, the "white space" can be filled with any amino acid (same as "+" shown in the generated rules).

The visualization program was implemented with the Python programming language. The use of the *matplotlib* plotting library allowed us to render an interactive 3D bar graph that displays a representation of the association rules. Functionality supported in this application includes zooming, rotating about any axis, and saving the current view of the graph as an image file.

4. Modified Rule Visualization and Results

4.1. Rule Visualization of Different Protein Classes

As shown in Table I, different rule sets are generated for different protein classes. The visualization of the top 30 association rules generated by the “all-β” class training set for the RS126 sets is displayed in Fig. 7. The top 10 association rules generated by the same class are shown in Fig. 6. The rules in Fig. 7 are sorted by secondary structure elements, position 5 (in the order of H, E, C, -), and then support value (maximum to minimum).



```
P,E,P,+,T,-,100,608,608,0.02956789
+,F,P,P,S,-,100,519,519,0.02523969
P,E,P,V,T,-,100,518,518,0.02519106
I,F,P,P,S,-,100,451,451,0.02193276
W,Y,+,Q,K,E,100,440,440,0.02139781
W,V,+,S,A,E,100,418,418,0.02032792
G,+,Y,T,K,E,100,416,416,0.02023066
+,W,Y,Q,Q,E,100,402,402,0.01954982
+,P,P,S,S,-,100,387,387,0.01882035
W,+,V,S,A,E,100,382,382,0.01857719
```

Fig. 6. The top 10 association rules generated by the “all-β” class training set for the RS126 set (rule sequence is different from Fig. 12).

It can clearly be seen that the rule sequences between Fig. 5 and Fig. 7 are different. Surprisingly, the top 30 “all-β” class rules do not produce all “E/Sheet” values at the decision attribute, position 5. In fact, some top rules have values “-/Others” at position 5. The top “all-β” class rules have similar support value as the top “all-α” class rules. It should be noted that Fig. 7 makes use of all 20 amino acids, compared to the 15 amino acids displayed in Fig. 6. The obvious different color distribution between the two diagrams indicates different rule value compositions. The manner in which different secondary structure elements are affected by their neighboring amino acid residues can be compared here, and users can zoom into the rules of interest to conduct a more detailed comparison and research (which is beyond the scope of this paper).

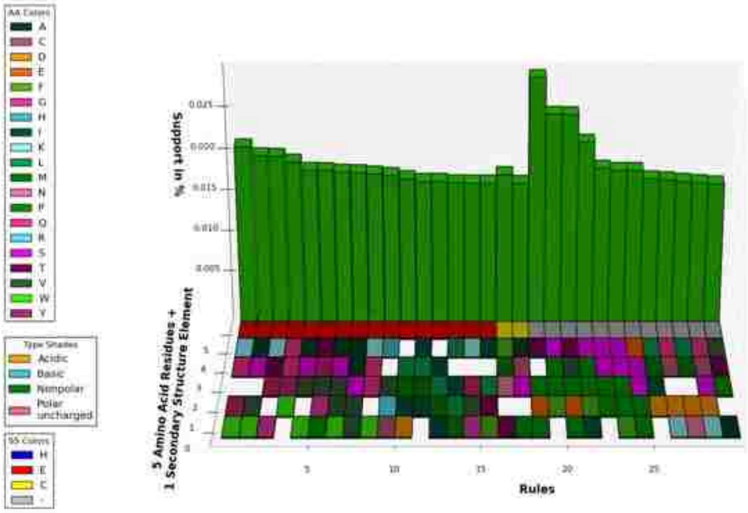


Fig. 7. Visualization of the top 30 association rules generated by the “all-β” class training set for the RS126 set (color by type).

Visualization of the top 30 association rules generated by the “α/β” class and “α+β” class training sets are shown in Fig. 8 and Fig. 9, respectively. It is interesting to note that although most values for both classes at position 5 are “H/Helix”, the amino acid values responsible for these values are quite different.

Visualization of antecedent association rules in a three-dimensional display allows patterns to emerge that would otherwise not be apparent. For example, in the graph for "all-α" by amino acid type in Fig. 5, it is apparent that acidic and basic amino acids occur at a frequency expected for the number of amino acids in those groups. Conversely, there is a significant preponderance of nonpolar amino acids and a paucity of polar uncharged. Also, it can be seen that although basic amino acids occur with expected frequency, overall they are concentrated in the middle position, 2, with fewer at both of the edge positions, 0 and 4. The preponderance of nonpolar amino acids is not equally distributed by position, and shows the inverse of the trend for basic amino acids (i.e., concentrated at the edge positions, 0 and 4, and fewer in the middle position, 2).

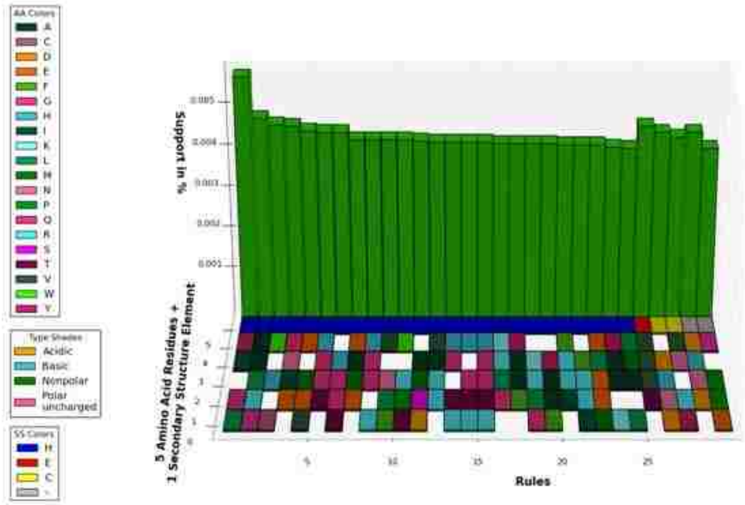


Fig. 8. Visualization of the top 30 association rules generated by the "α/β" class training set for the RS126 set (color by type).

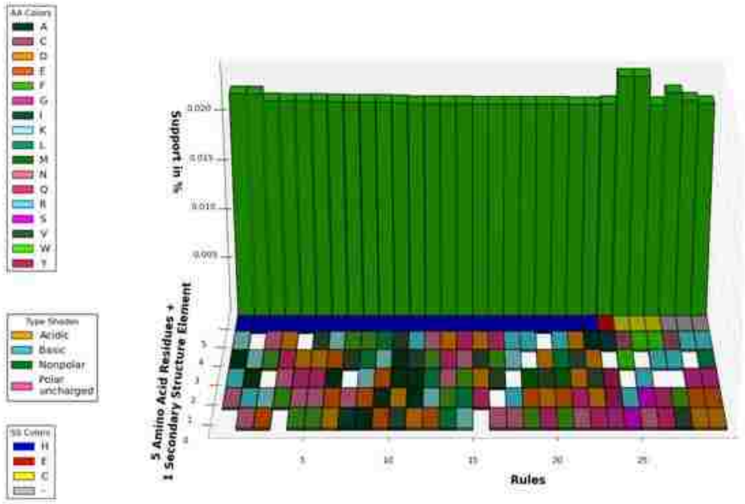


Fig. 9. Visualization of the top 30 association rules generated by "α+β" class training set for the RS126 set (color by type).

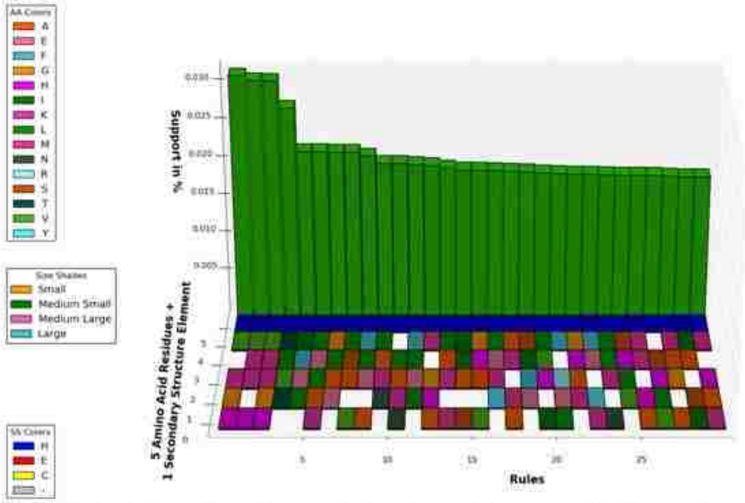


Fig. 10. Visualization of the top 30 association rules generated by "all-α" class training set for the RS126 set (color by size).

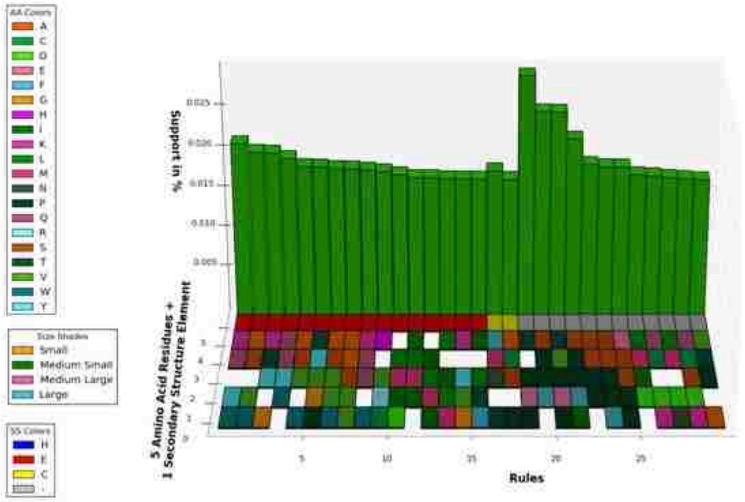


Fig. 11. Visualization of the top 30 association rules generated by "all-β" class training set for the RS126 set (color by size).

Similar patterns emerge from the graph for " all-α" by amino acid size, where amino acids were sorted by molecular weight into four groups (as shown in Fig.10, small: orange; medium small: green; medium large: pink; large: teal). There are significantly fewer amino acids of the large class, roughly the expected number of medium large and medium small, but significantly more than expected of the small class; here "expected" means that the amino acids occur at a frequency projected for the total number of amino acids in those weight groups. Among the medium large, the amino acids in this class are concentrated in the middle position, 2, and are less abundant in the edge positions, 0 and 4.

Comparison of the graphs between proteins classes also reveals patterns that are not apparent without visualization. Whereas the acidic and nonpolar amino acid types were roughly as abundant as expected, the basic and polar types were significantly different for the two protein classes. The basic amino acids are more numerous than expected in the “all-α" group, as compared to what was expected in the “all-β” group. The polar amino acids appear to be more abundant than expected in the “all-β” group, compared to what was expected in the “all-α" group; again, here “expected” means that the amino acids occur at a frequency projected for the total number of amino acids in those groups. Also, it becomes apparent that among the nonpolar type, different amino acids predominate in the “all-α" group versus the “all-β” group.

4.2. Rule Visualization of Different Test Proteins

The BLAST-RT-RICO prediction method uses the BLAST search to find a list of proteins with significant sequence alignments (for each test protein). Rules are generated from these proteins, and used for secondary structure prediction. Using the visualization technique, we can more readily get a sense of the information that the rules convey, and we can compare rule sets (generated by BLAST-RT-RICO) for test proteins. Proteins with significant sequence alignments may carry important evolutionary information, which can be captured statistically as rules for different test proteins.

Fig. 12 and Fig. 13 help us visualize the concept that different sets of amino acids are responsible for the two rule sets. The decision attribute values at position 5 for test protein A are all “E/Sheet” and “C/Coil.” The decision attribute values at position 5 for test protein B are mostly “H/Helix”, although all other possible values exist. The test protein A rules involve fewer amino acid positions compared to test protein B; as a result, there are more “gaps” comparatively. Perhaps, due to fact that test protein B involves more amino acid positions, the support values of the test protein B rules are comparatively lower than those for the test protein A rules. Because of the large number of rules, such visualization and comparisons would not have been feasible using only text-based representations of the rules.

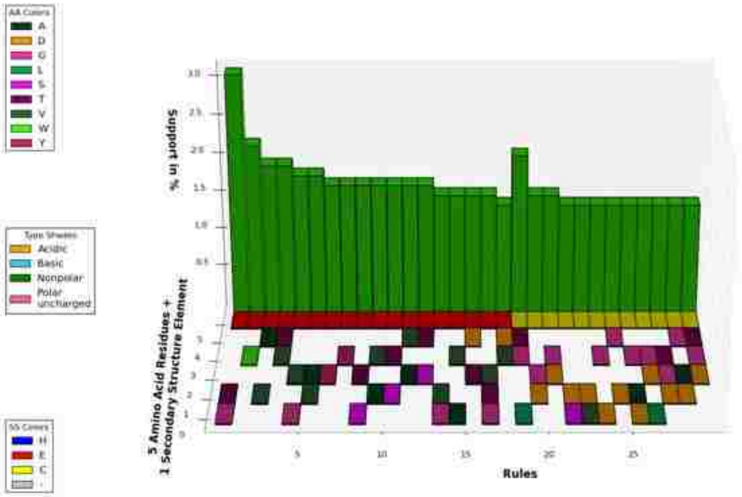


Fig. 12. Visualization of the top 30 association rules generated by test protein A (from RS126 set) using BLAST-RT-RICO (color by type).

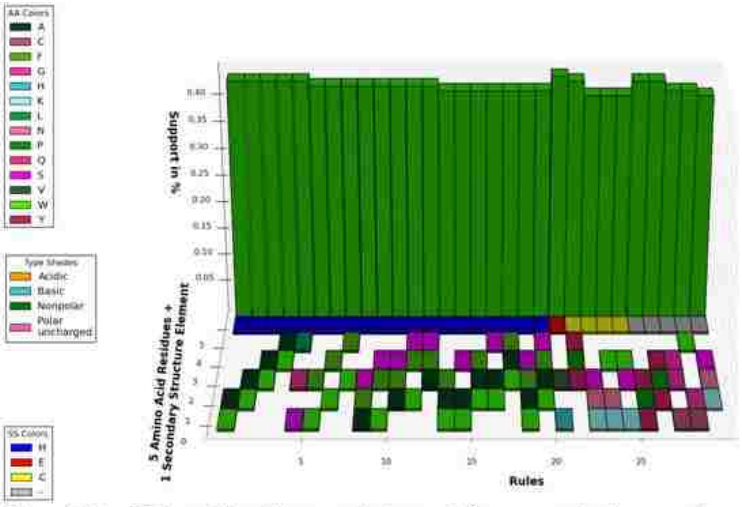


Fig. 13. Visualization of top 30 association rules generated by test protein B (from RS126 set) using BLAST-RT-RICO (color by type).

We have illustrated the value of visualization of antecedent association rules in a three-dimensional display with somewhat simple differences between the chemistry and size of amino acids. This rule visualization and comparison technique may lead to other future research topics related to protein secondary structure; for example, it encourages the researcher to ask questions such as: (1) how different rules (or groups of rules) affect the functions of an individual protein or a protein family, (2) why certain rules only exist in one protein class, but not in another, and (3) why some test proteins produce common rules although the proteins have different structure. In general, we believe that this approach will help researchers discern patterns of residue association in protein structure as other more complex properties of those amino acids are applied to the visualization.

5. Conclusions and Future Research

It is known that segment statistics can affect the accuracy of protein secondary structure prediction methods; that is, there are some relationships between secondary structure elements and their neighboring amino acid residues. RT-RICO and BLAST-RT-RICO are rule-based data mining methods that can be used to predict the secondary structure of proteins. The high Q_3 scores achieved by these methods support the validity of the generated rules. However, because of the large number of rules generated, potentially useful information within the rule sets had been difficult to identify. In this paper we presented a technique that not only enabled us to visualize those rules, but also allowed us to compare rule sets between different protein classes, and to compare rule sets of different test proteins.

For brevity, the figures in this paper each show only about 30 rules. On a twenty-one inch monitor, thousands of rules can be displayed and analyzed. Our software implementation supports features such as zooming and rotating, which allows users to have a “big picture” of a particular set of rules. For future research, it will be valuable to enhance this approach. For example, the user should be able to select groups of rules from the 3D display, and create a summary of statistics for analysis. It also might be possible to better understand the physio-chemical basis of structure by aligning similar rules together, and to examine rules in which some of the amino acids are the same, but the prediction is different.

In conclusion, we believe that such visualization provides additional value to the RT-RICO and BLAST-RT-RICO approaches for predicting protein secondary structure, providing much more insight than simply an accuracy score for the predictions.

References

Andreeva, A., Howorth, D., Chandonia, J. M., Brenner, S. E., Hubbard, T. J., Chothia, C. and Murzin, A. G., 2008, "Data growth and its impact on the SCOP database: new developments," *Nucleic Acids Res*, Vol. 36 (Database issue), D419-425.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F. and Nielsen, H., 2000, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, Vol. 16, No. 5, pp.412-424.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E., 2000, "The Protein Data Bank," *Nucleic Acids Res*., Vol. 28(1), pp. 235-242.

BLAST, 2009, "BLAST: Basic Local Alignment Search Tool," Obtained through the Internet: http://blast.ncbi.nlm.nih.gov/, [accessed 30/11/2009]

Cuff, J. A., and Barton, G., 1999, " Evaluation and improvement of multiple sequence methods for protein secondary structure prediction," *Proteins,* Vol. 34, pp. 508–519.

Fadime, U. Y., O¨zlem, Y., and Metin, T., 2008, "Prediction of secondary structures of proteinsnext term using a two-stage method," *Computers & Chemical Engineering*, Vol. 32(1-2), pp. 78-88.

Frishman, D., and Argos, P., 1997, "Seventy-five percent accuracy in protein secondary structure prediction," *Proteins,* Vol. 27, pp. 329–335.

Han, J. and Kamber, M., 2001, *Data Mining: Concepts and Techniques*, Morgan Kaufmann.

King, R. D., and Sternberg, M. J. E., 1996, "Identification and application of the concepts important for accurate and reliable protein secondary structure prediction," *Protein. Sci.*, Vol. 5, pp. 2298–2310.

Klepeis, J. L. and Floudas, C. A., 2002, "Ab initio prediction of helical segments in polypeptides," *J Comput. Chem*, Vol. 23, No. 2, pp.245-266.

Lee, L., Leopold, J. L., Kandoth, C. , and Frank, R. L., 2010a, "Protein secondary structure prediction using RT-RICO: a rule-based approach," *The Open Bioinformatics Journal*, accepted for publication.

Lee, L., Leopold, J. L., and Frank, R. L., 2010b, "Protein secondary structure prediction using BLAST and Relaxed Threshold Rule Induction from Coverings," *BMC Bioinformatics*, under review.

Levitt, M. and Chothia, C., 1976, "Structural patterns in globular proteins," *Nature*, Vol. 261, No. 5561, pp.552-558.

Murzin, A. G., Brenner, S. E., Hubbard, T. and Chothia, C, 1995, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *J Mol. Biol,* Vol. 247, No. 4, pp.536-540.

Rost, B., and Sander, C., 1993, “Prediction of protein secondary structure at better than 70% accuracy,” *J. Mol. Biol.*, Vol. 232, pp. 584-599.

Rost, B., 2003, “Rising accuracy of protein secondary structure prediction,” in: Chasman, D. (Ed.), *Protein structure determination, analysis, and modeling for drug discovery.*, New York: Dekker, pp. 207–249.

Salamov, A. A., and Solovyev, V. V., 1995, “Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments,” *J Mol. Biol.*, Vol. 247, pp. 11–15.

Wong, P. C., Whitney, P., and Thomas, J., 1999, “Visualizing Association Rules for Text Mining,” *Proceedings of the 1999 IEEE Symposium on Information Visualization*, pp. 120-123, 152.

Zhang, C. T. and Zhang, R., 2003, “Q9, a content-balancing accuracy index to evaluate algorithms of protein secondary structure prediction,” *Int J Biochem Cell Biol.*, Vol. 35, No. 8, pp.1256-1262.

SECTION

3. CONCLUSIONS

A novel rule-based method, RT-RICO, which generates rules that can be used in predicting protein secondary structure, was presented in this dissertation. Rule-based RT-RICO (discussed in paper 3) achieved the Q_3 accuracy scores of 81.75% for the RS126 set and 79.19% for the CB396 set. The BLAST-RT-RICO approach (discussed in paper 4), which utilizes data from proteins with significant sequence alignments, attained the Q_3 scores of 89.93% for the RS126 set and 87.71% for the CB396 set. These scores are better than the Q_3 scores that have been reported for comparable computational methods using the same datasets.

The main RT-RICO rule generation algorithm has a time complexity of $O(m^2 2^n)$, with m^2 dominating the time complexity. The current implementation of the algorithm enables the generation of rules from the available protein data within an acceptable timeframe, resulting in efficient prediction of the secondary structure of available test datasets.

Because of the large number of rules generated by RT-RICO and BLAST-RT-RICO, potentially useful information within the rule sets can be difficult to identify. Paper 5, Rule Visualization, presented a technique that not only enabled us to visualize those rules, but also allowed us to compare rule sets between different protein classes, and to compare rule sets of different test proteins.

In the future, the next natural step would be to construct a BLAST-RT-RICO prediction server with functions to analyze training datasets and prediction results. A server implementation also would make this promising rule-based prediction method more easily accessible to the broader research community.

BIBLIOGRAPHY

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., and Nielsen, H. (2000) 'Assessing the accuracy of prediction algorithms for classification: an overview,' *Bioinformatics*, Vol. 16, No. 5, pp. 412-424.

BLAST (2009). BLAST: Basic Local Alignment Search Tool. Obtained through the Internet: http://blast.ncbi.nlm.nih.gov/, [accessed 11/30/2009].

Cuff, J. A. and Barton, G. (1999) 'Evaluation and improvement of multiple sequence methods for protein secondary structure prediction,' *Proteins*, Vol. 34, pp.508–519.

Fadime, U. Y., O¨zlem, Y., and Metin, T. (2008) 'Prediction of secondary structures of proteinsnext term using a two-stage method,' *Computers & Chemical Engineering*, Vol. 32, No. 1-2, pp. 78-88.

Frishman, D. and Argos, P. (1997) 'Seventy-five percent accuracy in protein secondary structure prediction,' *Proteins*, Vol. 27, pp.329–335.

Grzymala-Busse, J. W. (1991) *Managing Uncertanity in Expert System,* Boston: Kluwer Academic.

Hu, H., Pan, Y., Harrison, R. and Tai, P. (2004) 'Improved protein secondary structure prediction using support vector machine and a new encoding scheme and an advanced tertiary classifier,' *IEEE Trans NanoBiosci*, Vol. 3, pp.265–271.

Jones, D. T. (1999) 'Protein secondary structure prediction based on position-specific scoring matrices,' *J Mol Biol*, Vol. 292, No. 2, pp.195-202.

Kabsch, W., and Sander, C. (1983) 'Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features,' *Biopolymers*, Vol. 22, No. 12, pp. 2577–2637.

Kim, H. and Park, H., (2003) 'Protein secondary structure prediction based on an improved support vector machines approach,' *Protein Eng*, Vol. 16, pp.553-60.

King, R. D. and Sternberg, M. J. E. (1996) 'Identification and application of the concepts important for accurate and reliable protein secondary structure prediction,' *Protein Sci*, Vol. 5, pp.2298–2310.

Lee, L., Leopold, J. L., Frank, R. L., and Maglia, A. M. (2009) 'Protein Secondary Structure Prediction Using Rule Induction from Coverings,' *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology 2009*, Nashville, Tennessee, USA, pp. 79-86.

Lee, L., Kandoth, C., Leopold, J. L., and Frank, R. L. (2010a) ‘Protein Secondary Structure Prediction Using Parallelized Rule Induction from Coverings,’ *International Journal of Medicine and Medical Sciences,* Vol. 1, No. 2, pp. 99-105.

Lee, L., Leopold, J. L., Kandoth, C., and Frank, R. L. (2010b) ‘Protein secondary structure prediction using RT-RICO: a rule-based approach,’ *The Open Bioinformatics Journal,* Accepted for publication.

Lee, L., Leopold, J. L., and Frank, R. L. (2010c) ‘Protein secondary structure prediction using BLAST and Relaxed Threshold Rule Induction from Coverings,’ *BMC Bioinformatics,* in review.

Lee, L., Leopold, J. L., Edgett, P. G., and Frank, R. L. (2010d) ‘Rule Visualization of Protein Motif Sequence Data for Secondary Structure Prediction,’ *Proceedings of ANNIE 2010 conference,* St. Louis, Missouri, USA.

Nguyen, N. and Rajapakse, J. C. (2007) ‘Two stage support vector machines for protein secondary structure prediction,’ *Intl J Data Mining & Bioinformatics*, Vol. 1, pp.248-269.

Pawlak, Z. (1984) ‘Rough Classification,’ *Int J Man-Machine Studies,* Vol. 20, pp.469-483.

Rost, B. (2003) ‘Rising accuracy of protein secondary structure prediction,’ *In:* Chasman, D. (ed.), *Protein structure determination, analysis, and modeling for drug discovery,* (pp. 207–249), New York: Dekker.

Rost, B., and Sander, C. (1993a) ‘Prediction of protein secondary structure at better than 70% accuracy,’ *J Mol Biol,*Vol. 232, pp. 584-599.

Rost, B., and Sander, C. (1993b) ‘Improved prediction of protein secondary structure by use of sequence profiles and neural networks,’ *Proc Natl Acad Sci USA*, Vol. 90, pp. 7558–7562.

Salamov, A. A. and Solovyev, V. V. (1995) ‘Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments,’ *J Mol Biol*, Vol. 247, pp.11–15.

Zhang, C. T., and Zhang, R. (2003) ‘Q9, a content-balancing accuracy index to evaluate algorithms of protein secondary structure prediction,’ *Int J Biochem Cell Biol,* Vol. 35, No. 8, pp. 1256-1262.

VITA

Leong Lee attended St. Paul’s College, Hong Kong; Raffles Institution, Singapore and Raffles Junior College, Singapore. He received the degree of Bachelor of Science (Computer and Information Sciences) from the National University of Singapore. During the following two years he was employed as a system engineer at Shell Eastern Petroleum, Singapore. Leong Lee worked in Temasek Polytechnic, Singapore for seven years, as an IT lecturer / course coordinator. He also received a Master of Science (Information Management) degree from the National University of Ireland, Dublin.

In August, 2006, he joined Missouri University of Science and Technology (formerly University of Missouri-Rolla) as a graduate student. On December 2007, he received his Master of Science in Computer Science degree. Leong Lee then pursued the Doctor of Philosophy degree in Computer Science. He will receive his Ph.D. from the Department of Computer Science at Missouri University of Science and Technology in December 2010.