International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:12, 2009

Protein Secondary Structure Prediction Using Parallelized Rule Induction from Coverings

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

***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

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)

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.

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**

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.

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:

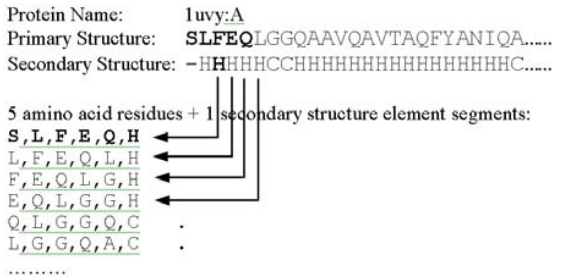
(G, H, I) => Helix H
(E, B) => Sheet E
(T, S) => Coil C
(-) => "-"

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).



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 $m \times (n+1)$ 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].

```
+,+,+,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 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%.

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%.

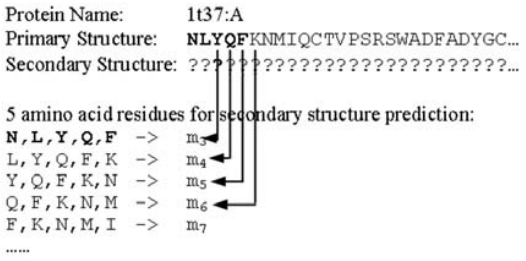


Fig. 3. Protein primary structure 5-residue segments and related secondary structure elements prediction. m_i is an element of set {H,E,C,-}. It is then converted to an element of the set {H, E, C}. Note: The first and second positions at the beginning of the sequence are represented (predicted) by 3 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 $P_k = \{\{x_{i1}, x_{i2}, \ldots, x_{ik}\} \mid x_{i1}, x_{i2}, \ldots, x_{ik} \in S\}$.

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 (∏_{x∈P} [x]* ≤ 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.

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: $4 \times 20^n \times (2^n\text{-}1)$ i.e. $\mathbf{O(20^n \times 2^n)}$

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

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).

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.

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**

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]).

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.