String-Based Models for Predicting RNA-Protein Interaction

Donald Adjeroh, Maen Allaga, Jun Tan
West Virginia University
donald.adjeroh@mail.wvu.edu

Jie Lin, Yue Jiang
Fujian Normal University
Fuzhou, China

Ahmed Abbasi
University of Virginia

Xiaobo Zhou
Wake Forest University
Health Sciences

ABSTRACT

In this work, we study string-based approaches for the problem of RNA-Protein Interaction (RPI). We apply string algorithms and data structures to extract effective string patterns for prediction of RPI, using both sequence information (protein and RNA sequences), and structure information (protein and RNA secondary structures). This led to different string-based models for predicting interacting RNA-protein pairs. We show results that demonstrate the effectiveness of the proposed string-based models, including comparative results against state-of-the-art methods.

General Terms

Algorithms, Theory

Keywords

RNA protein interaction; RPI; *k*-mers; suffix trees; richness.

1. INTRODUCTION

The interaction between proteins and RNA is known to be an important cellular process. RNA interaction with malfunctioning proteins has been implicated in cell misregulation, leading to serious diseases [1, 3]. Yet, we still lack a complete understanding of the characteristics of a protein or of an RNA that allow them to interact. Even less is known about those characteristics that play a significant role in the formation of new protein-RNA complexes. Various groups have thus studied interactions for specific pairs of protein and RNA, even without knowing the general properties that facilitate or inhibit such interactions. Starting with experimentally known interacting RNA-protein pairs, computational methods, such as machine learning can be brought to bear on the problem, by attempting to predict possible RNA-protein interactions based on information from the already known interacting RNAs and proteins [2, 3].

The protein or RNA secondary structure describes how the molecules are bound together in a three dimensional space, and therefore can play a crucial role in characterizing the interaction process. The RNA secondary structure describes how some nucleotides in the single stranded RNA are paired to form potentially complicated structures, such as stems, loops, and hairpins. Some methods have been developed to predict RNA secondary structure based on the nucleotide sequences. Thermodynamic methods are the most popular amongst these methods [4, 5]. These mainly rely on the notion of free energy, and building secondary structures based on the minimum free energy principle [4]. On the other hand, protein secondary structures describe how the amino acids are positioned in a three dimensional space. Various approaches have been used to describe the protein secondary structure. A popular approach is by using the dihedral angles (φ and ψ) between the amino acids. Dihedral angles define the angles of rotation between two planes, in this case, the planes are defined by the bonds between three adjacent amino acids [6]. Ramachandran codes are derived from Ramachandran plots (which are 2D charts describing the distribution of the dihedral angles), by reducing information in the plot to some clusters [6, 7, 8]. Another approach is the protein blocks, which describe protein secondary structures based on the folds formed by five consecutive amino acids and then clustering these folds [9].

Earlier methods for RNA-Protein interaction prediction focused on sequence data, building features based on only sequence information, for RNA and protein individually, or combining both sequences to extract representative features [3]. Recently, secondary structures were shown to be important in the interaction process, and are now being included in the prediction [10]. In this work, we consider different representations for proteins and RNAs. We consider sequence and structural information for both protein and RNA. For the sequences, we use the traditional 4-letter alphabet for RNA (A, C, G, U), but a 7-letter reduced alphabet for protein. For structures, we use the Ramachandran codes for protein structure representation. These have been used in previous work on studying protein structures [6, 7]. However, this work represents the first time Ramachandran codes are being used for protein-RNA interaction studies. RNA secondary structures have been

ACM-BCB '17, August 20-23, 2017, Boston, MA, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4722-8/17/08…$15.00
http://dx.doi.org/10.1145/3107411.3107508

used earlier in RNA alignment and Protein-RNA interaction prediction. In this work, we consider string-based representations for both the RNA secondary structure, and the protein secondary structure, in addition to the traditional protein and RNA sequences.

Armed with these representations, we build two different prediction models using a string-based approach. Here, we built a feature space based on *k*-grams (i.e., *k*-length substrings, also called *k*-mers in the biology literature). We analyze the sequences of Protein-RNA pairs to determine the *k*-mers that tend to appear in interacting pairs, and those that are often found in non-interacting pairs. We then used these *k*-mer pairs as our descriptors (feature set) for predicting RNA-protein interaction. We applied this approach on both sequence-based information and structure-based information, after representing the structural information as strings.

The reminder of this paper is organized as follows: Section 2 provides some background, and discusses previous work on predicting Protein-RNA interaction. Section 3 describes the proposed string-based approach. Section 4 presents the results, including comparison with the state-of-the-art. Section 5 concludes the paper.

2. BACKGROUND & RELATED WORK

The importance of RNA-Protein interaction comes from the key role it plays in regulating cellular processes. Researches have shown interest in RNA-Protein interactions primarily driven by the need to understand how cells work, including cell localization and other fundamental processes [10]. Studies have shown that some cases of RNA-Protein interaction are related to some important diseases [1]. Clearly, the problem of prediction is closely related the issue of representing the RNAs and proteins. Success in identifying and extracting the information that is most relevant to protein-RNA interaction will no doubt lead to improved computational prediction of such interactions.

Methods for addressing the RNA-Protein Interaction (RPI) problem can be traced to those that have been used to study the related problem of protein-protein interaction (PPI). The Protein-Protein interaction problem is well studied due to the importance of proteins in all cell processes, including coding and decoding genes. Shen et. al [11] were among the first to predict protein-protein interaction using only sequence information. They used a simplified 7-character alphabet to represent protein sequences and built a Support Vector Machine (SVM) prediction model using a high quality database containing 16,443 experimentally validated entries. They were not only able to predict Protein-Protein interaction, but also to build a protein interaction network that shows the relationships and connections between different proteins based on their interactions [11]. For other related work on protein-protein interaction see [12, 13, 14].

Muppirala et. al. [3] explored RNA and protein sequences separately. They build a 599-dimensional feature space, with 343 features extracted from protein and 256 features from RNA. Similar to Shen et. al. [11], the 343 protein features were extracted by first considering the 7-class reduced protein alphabet, whereby the 20 amino-acids are clustered into 7 groups based on their dipole moments and their side-chain volumes. To conserve locality information, the notion of triads was used to extend the feature space to 7x7x7 features for protein, and 4x4x4 features for RNA. Two classification models (SVM and Random Forests) were deployed to build the prediction scheme. The models were trained using two datasets, namely, RPI369 and RPI2241. The Random Forest model trained with RPI2241 obtained the most accurate results among other trained models, achieving 89.6% in accuracy, with 0.89 and 0.90 for precision and recall, respectively.

In [15], Wang et. al. applied *k*-mer approach by finding the pairs of protein amino acids and RNA nucleotides that tends to appear together. They worked with a reduced protein alphabet, the 20 amino acids were grouped based on their charge and polarity. The new alphabet consisted of 4 groups representing the 20 amino acids. Then, they considered protein 4-mers and RNA 3-mers. This allowed them to preserve some locality information, which was indicated to have high impact on the prediction process. The feature space consisted of a 4094-dimensional space (4^3 features for protein, and 4^3 features for RNA). This high dimensional space requires relatively large datasets for training. Thus, they selected only 500 features that showed the highest impact on the prediction, and adopted the naive-Bayes classifier as the basic classification method. They tested the method on different datasets, including RPI369, RPI2241 and NPInter. The accuracy achieved for these three datasets were 75%, 74% and 77.6%, respectively.

RPI-Pred [10] was developed by Suresh et. al. to predict the interaction between non-coding RNA and proteins. They included information from both sequences and secondary structures, building a features-vector of 132 features. Here, they used 20 features to describe the RNA considering 4 different nucleotides and 5 secondary structure elements including stem, hairpin, loop, bulges. They used 112 features to characterize proteins, following the reduced 7-class alphabet for amino acids, and the extracted 16 protein blocks [9]. They introduced a new dataset, namely, RPI1807 and used this for training. They tested their method using three datasets, namely, RPI369, RPI2241 and NPInter. Using SVM as the prediction scheme, they achieved an accuracy of 92%, 84% and 86.9% using RPI369, RPI2241 and NPInter, respectively.

Lu et. al. [16] followed a different approach in predicting RNA-Protein interaction. The core difference in their work is in how they extracted their features. RNA secondary structures were predicted using Vienna RNA package. Additionally, they used hydrogen bonding information, then using the Fourier transform they extracted a feature set for both RNA and protein to form the feature vector for each RNA and protein. They included the first ten terms of the Fourier series for each information type. They built a training dataset containing 649 non-redundant protein-RNA pairs (322 of these were interacting pairs, and the remaining 327 were non-interactive pairs). They trained a scoring matrix which gives a score for each protein-RNA pair. Based on the assigned score, they predict the interaction between protein and RNA. The method achieved a 77% accuracy on the NPInter dataset.

Recently, the problem of RNA-Protein interaction has been considered from the viewpoint of complete structural representations. Zhang et al. [17] developed a deep learning model to define the preferences of RNA Binding Protein (RBP) structural representations. They used information from predicted RNA tertiary structures to study the problem of RNA-Protein interaction. This helped them to define a 3D representation for RPI complexes, which were then used to describe the binding preferences.

For both RNA and protein, their sequences and secondary structures can each be expressed in the form of strings [18, 19]. In this work, we take advantage of efficient string algorithms and data structures to extract discriminative feature sets and build prediction models that can predict RNA-Protein interaction. The other contribution is the development of a prediction model built on Ramachandran codes for protein structures, and enhanced descriptors of RNA secondary structure elements. In this work, we used both Support Vector Machines (SVM) and Random Forests (RF) to explore the influence of classification schemes on the prediction performance.

3. STRING-BASED MODELS

In this work, we will use information from both sequences and secondary structures. Before we discuss our string-based approach, we first describe how the information from the protein and RNAs are represented.

3.1 Representing RNAs

For RNA sequences, although a nucleotide can be any of A, U, C and G nucleotides, some RNAs are not completely known and they include X at some positions denoting that the nucleotide at this position is unknown, therefore the alphabet for RNA sequences is extended to 5 characters (A, U, C, G, X). This alphabet will be used to encode RNA sequences in this work. We also use information from the RNA secondary structure. In this work, when the RNA structure is unknown, we will use RNAFold (part of the Vienna RNA suite of programs [4, 20]) to predict the structure. The method implements a free energy model to predict the secondary structure for a given RNA sequence. To describe the RNA secondary structure, we use the RNA secondary structure elements (SSE) -- see Figure 1 (taken from [18]). For simplicity, we consider only the five basic types, namely, Single strands, Stem (or stack), Loop, Internal Loop and Bulges. The RNA secondary structure is thus represented as a string -- sequence of basic SSEs.

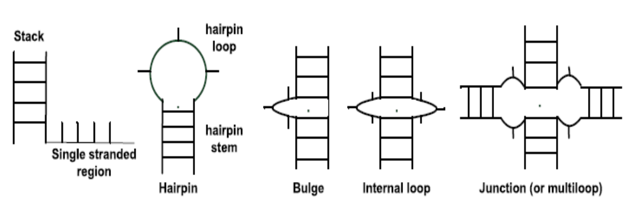

Figure 1: RNA secondary structure elements [18].

3.2 Representing Protein

Given the chemical similarity between amino acids, we can group amino acids into different groups and use the groups, rather than the individual amino acids, to represent protein sequences. Grouping the amino acids can be done based on various criteria. In this work, we follow [3, 10], where amino acids were classified based on their dipole moments and their side chain volumes. This way the original 20 amino acids are be classified into 7 groups: I: {Ala, Gly, Val}, II: {Ile, Leu, Phe, Pro}, III: {Tyr, Met, Thr, Ser}, IV: {His, Asn, Gln, Trp}, V: {Arg, Lys}, VI: {Asp, Glu} and VII: {Cys}. The protein secondary structure is defined by the positions of consecutive molecules in 3D space. The positions can be described by the dihedral angles between each three consecutive molecules, denoted Omega (ω), phi (φ) and psi (ψ). Due to the limitation in ω angle, usually only φ and ψ angles are considered in describing the protein secondary structure. Ramachandran et. al. [21] studied the relationship between φ and ψ and presented 2D plots (now called Ramachandran plots), showing the density of the joint occurrences of these angles. This density can be used to derive protein secondary structure representations by clustering the values from the Ramachandran plot, then replacing the values of (φ, ψ) pairs by their cluster representative. To represent protein structure, Suresh et al [10] used 16-character protein blocks. In this work, we will use a 7-symbol alphabet representing 7 clusters from the Ramachandran plot. See [6, 7]. Hence, the protein representation used will consist of a 7-character alphabet for amino acid groups (for the sequence), and another 7-character alphabet from the Ramachandran codes (for the protein secondary structure).

3.3 String–based Approach

Although the feature-based approach is powerful and contains a lot of information about RNA and protein, it does disregard important local information. This local information could be important in the prediction process. When an RNA bonds to a protein, it is not just an amino acid and a nucleotide that are involved, but a set of neighboring nucleotides against a set of amino acids.

Furthermore, the secondary structures should be compatible to allow RNA-Protein bonding. These observations suggest that we could consider small portions or *k*-grams (*k*-mers) of sequences and structures when building the feature vector rather than looking for individual molecules. This will typically lead to a very large dimensional feature space. Consider for instance, the 5-mer strings under a 25-character alphabet (using the RNA representation presented earlier). This means we have more than $25^5 \approx 9.8\times10^6$ possible different 5-mers to consider for RNA only, besides the $49^5 \approx 2\times10^8$ 5-mers for proteins. This is a huge feature space that will be very difficult to handle with current computational limitations. Thus, we need to reduce the feature space dimensionality. A quick observation is that this feature space will be very sparse, as most combinations of the RNA and protein symbols will not occur in practice.

Reducing feature space dimensionality means we need to carefully select some *k*-mers and drop the rest. Not all *k*-mer strings hold the same amount of information. Thus, to enhance prediction accuracy, we need to identify the *k*-mers that have the most influence on the prediction of interacting or non-interacting RNA-protein pairs. Hence, we look for the *k*-mer strings that appear mostly in the positive pairs and those that appear mostly in the negative pairs, and then construct a feature vector based on these.

3.4 Suffix trees for protein and RNA strings

The naive approach to find the most occurring substrings is to count each of them within the dataset. The running time for this approach will be O(nk) for *each k*-mer, where n is the total number of all characters in the dataset. This means we need to hold a large dictionary of all possible substrings, the size of this dictionary would be in $O(\alpha^k)$, where α is the alphabet size (in our case, $\alpha=25$ for RNA and α=49 for protein). We need to find a better approach to study the distribution of *k*-mers, i.e. a memory- and time-efficient method to find the *k*-mers that contribute most in the interaction process. An improved approach will be to go over the database to first determine all the *k*-mers that actually occurred, and then use standard linear-time pattern matching algorithms to determine their respective number of occurrences. Overall, this will be $O(kn^2)$ time worst case.

A better approach will be to use suffix trees and suffix arrays [19, 22-25] to provide a better tool to find the most occurring substrings within positive pairs and negative. The suffix tree requires an $O(n)$ time and space for construction. After construction, we can traverse the $O(n)$ nodes of the suffix tree to determine the occurrence counts of all substrings in $O(n)$ time. Thus, this is the time required to count all the *k*-mers in the string, independent of *k*. The power in using the suffix tree to find the distribution of *k*-mers is that we don't need to maintain a large dictionary. We built suffix trees counting occurrences of each substrings of length 2 to 5 for RNA sequence and secondary structure, and protein sequence and secondary structure. That is, we constructed four suffix trees, one for each type of string representation we used, namely: RNA sequence, RNA secondary structure elements represented as strings, protein sequence, protein secondary structure represented as a sequence of Ramachandran codes.

3.5 Richness for protein and RNA substrings

In general, the *k*-mers that tend to occur more in positive pairs (i.e., interacting RNA-protein pairs) would provide more information in deciding on a positive pair than other *k*-mers that appeared equally in both positive and negative pairs. Similarly for *k*-mers that appeared more in negative pairs. With this observation, we should look for more than just occurrence counts. The richness could be a better measure of the contribution of a *k*-mer to the interaction between RNA-protein pairs. Given a *k*-mer, say β, let:

$\gamma_+(\beta)$ = #occurrences of β in the positive pairs.
$\gamma_-(\beta)$ = #occurrences of β in the negative pairs.

The *k*-mer richness is simply defined as:

$R(\beta) = (\gamma_+(\beta)+1)/(\gamma_-(\beta)+1)$.

Thus, *k*-mers with richness greater than 1 appear more in positive pairs (positive *k*-mers), while a richness value near zero means the *k*-mer appeared mostly in the negative pairs (negative *k*-mer). Richness values close to 1 are associated with *k*-mers that appear equally in both positive and negative pairs, hence they provide less discrimination ability between interacting and non-interacting pairs.

Given the foregoing, we now construct four suffix trees for positive dataset, and another four suffix trees for the negative dataset. We extracted pairs that appeared only in positive pairs or only in negative pairs as they hold the most discriminative information for interaction prediction. To provide some perspective on the feature space generated, we used the RPI1807 dataset (see Section 4.1 on datasets), to build suffix trees of depth five and computed the distributions for *k*-mers up to length 5. RNA 4-mers appeared for as many as 18,020 times in positive pairs and 16,894 in negative pairs, while for 5-mers, the number drops to 3,700 appearances in positive pairs and 3,494 negative pairs. Obviously, the numbers drop as the *k*-mer length increases. Protein 4-mers occurred about 1,134 times in positive pairs, and 443 times in negative pairs. As mentioned, the direct *k*-mer count distribution may not be the best way to capture the information carried by the *k*-mers. Thus, we considered the richness as the main factor to compare *k*-mers and build the feature vector. Figure 2 shows the RNA *k*-mer richness (log scale) for the RPI1807 dataset above. In this chart, positive log values imply *k*-mers that appear more in the positive pairs, while negative values correspond to those that appear more in the negative pairs. Thus, a simple threshold can be used to select the positive *k*-mers and the negative *k*-mers, while avoiding those that are less discriminative (those close to 0).

3.6 String-based Models

Based on the foregoing, we then consider string-based prediction models that combine the extracted RNA and protein *k*-mers. Our models consider different combinations of the identified *k*-mers that pass the richness threshold. We consider 5 models based on these combinations, viz: QQ: RNA sequence *k*-mers in combination with protein sequence *k*-mers; SS: RNA secondary structure *k*-mers and protein secondary structure *k*-mers, QS: RNA sequence *k*-mers and protein secondary structure *k*-mers, and SQ: RNA secondary structure *k*-mers and protein sequence *k*-mers. The final model (QSQS) combines *k*-mers from the first four models, thus exploiting info from RNA sequence, and RNA structure, protein sequence, and protein structure.

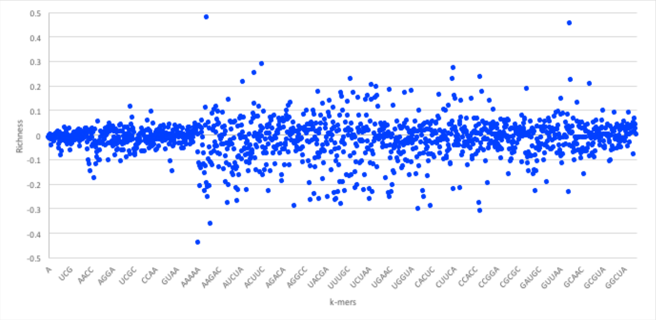

Figure 2: RNA sequence *k*-mer richness (log values) using the RPI1807 dataset. Not all *k*-mers are shown.

4. EXPERIMENTS & RESULTS

4.1 Datasets and Setup

We performed experiments to test the performance of the proposed methods, using some known datasets. Classification was performed using both SVM and Random Forests (RF) using Weka, version 3.6.13. SVM parameters were set as C=2^5, $\gamma=2^{-7}$ for the string-based models. For RF, the number of decision trees was set to 200. We used the RPI1807 dataset from [10] to construct our models, and set parameters. The dataset has 1,807 positive pairs (1807 protein and 1078 RNA chains), and 1436 negative pairs (including 1436 protein and 493 RNA chains). Then, we evaluated the models on the RPI369 and RPI2241 datasets reported in [3]. Both were obtained from the PRIDB dataset of RNA-protein complexes [26] extracted from the protein databank (PDB). RPI2241 includes complexes with rRNA, ncRNA and mRNA, and this is more challenging.

4.2 Performance Measurement

We evaluated our approaches using 10-fold cross-validation. To measure the performance, we used precision (PRE), recall (REC), accuracy (ACC), and F-measure (FSC), viz: PRE=TP/(TP+FP), REC=TP/(TP+FN), ACC=(TP+TN)/(TP+TN+FP+FN), FSC=2*(PRE*REC)/(PRE+REC), where, TP is true positive (the count of correctly classified positive pairs), FP is false positive (the count of wrongly classified positive pairs), TN is true negative (count of correctly classified negative pairs), and FN is false negative (count of wrongly classified negative pairs). We also computed the area under the curve (AUC) (with values in [0 1], with 1 indicating perfect prediction).

4.3 Results

Table 1 shows the results of our proposed string-based models. Columns 2 and 3 show results for the QSQS model, using both SVM and RF. Clearly, RF is doing much better than SVM using our approach. Thus, subsequent results in this work are reported only for the RF classifier.

Table 1: Results for the proposed string-based models

Metric	**QSQS (SVM)**	**QSQS (RF)**	**QQ**	**SS**	**QS**	**SQ**
#*k*-mers	7,030	7,030	4,680	2,350	3,255	2,955
AUC	0.75	0.98	0.93	0.66	0.64	0.95
PRE	0.79	0.93	0.93	0.79	0.78	0.91
REC	0.74	0.93	0.93	0.67	0.65	0.89
FSC	0.73	0.93	0.98	0.61	0.58	0.89
ACC (%)	74.00	93.35	93.19	66.98	65.15	89.29

As expected, the QSQS model achieved the best result. This is due to its use of more detailed information from both structure and sequence. The results of using only sequences (QQ model) were very close to using all available info. This could be due to the fact that secondary structure is determined by the sequence. Interestingly, using *k*-mers from only the secondary structures (SS model) led to a significant performance drop (ACC=66.98%). The table shows that sequences provide a key information for RPI prediction. Though secondary structures can help when combined with sequences, they lack precision when used independently.

4.4 Comparison with State-of-the-Art

We compared our models with two recent approaches reported in [10] and [3]. Table 2 shows the comparative results using the RPI2241 dataset. Our string-based model (QSQS) had an accuracy of 86.5%, performing better than RPI-Pred (84.0%) and close to RPISeq-SVM (87.1%).

Table 2: Comparative analysis of proposed string-based models on RPI2241 dataset

Metric	String-Based	RPI-Pred[10]	RPISeq-SVM[3]	RPISeq-RF [3]
AUC	0.92	0.89	0.97	0.92
PRE	0.86	0.88	0.87	0.89
REC	0.86	0.78	0.88	0.90
FSC	0.86	0.83	0.87	0.90
ACC (%)	86.5	84.0	87.1	89.6

Table 3 shows the results when using the RPI369 dataset. The string-based method outperformed all three competing methods (ACC= 96.38%). The proposed models were more consistent over different datasets. Our string-based had an accuracy of at least 86.52% over each dataset.

Table 3: Comparative analysis of proposed string-based models on RPI369 dataset

Metric	String-Based	RPI-Pred[10]	RPISeq-SVM[3]	RPISeq-RF [3]
AUC	0.98	0.95	0.81	0.81
PRE	0.96	0.89	0.73	0.75
REC	0.96	0.89	0.73	0.78
FSC	0.96	0.89	0.73	0.77
ACC (%)	96.38	92.0	72.8	76.2

5. CONCLUSION

We have introduced different string-based models for predicting RPI. We used the Ramachandran codes to represent protein secondary structure, and developed an innovative *k*-mers approach using powerful string data structures to address the problem of RPI prediction. The string-based approach maintained locality information which plays a key role in interaction between RNA and protein. Our approach showed comparable performance with the state-of-the-art methods on one dataset, while outperforming the methods on a second dataset.

Further improvement could be obtained by finding better ways to integrate the sequence and structure information, and intelligent fusion of the feature-based and string-based approaches. The string-based approach can also explore longer substrings (the *k*-mers), or perhaps inexact *k*-mers. Computing the *k*-mer richness by considering the sequence and structure information jointly, rather than separately, could lead to further improvements.

6. ACKNOWLEDGMENTS

Part of this work is supported by funds from the NSF (IIS-1552860, IIS-1553109), and from the NIH (1R01GM123037-01).

7. BIBLIOGRAPHY

[1] Jankowsky, E., & Harris, M. E. (2015). Specificity and nonspecificity in RNA–protein interactions. Nature Reviews Molecular Cell Biology, 16(9), 533–544.

[2] Khalil, A. M., & Rinn, J. L. (2011). RNA-protein interactions in human health and disease. Seminars in Cell & Developmental Biology, 22(4), 359–65.

[3] Muppirala, U. K., Honavar, V. G., & Dobbs, D. (2011). Predicting RNA-protein interactions using only sequence information. BMC Bioinformatics, 12(1), 489.

[4] Hofacker, I. L. (2003). Vienna RNA secondary structure server. Nucleic Acids Research, 31(13), 3429–31.

[5] Zuker, M., & Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Research, 9(1), 133–48.

[6] Lo, W.-C. et al. (2007). Protein structural similarity search by Ramachandran codes. BMC Bioinformatics, 8, 307.

[7] Tan J. & Adjeroh D. (2015). Text encoding for protein structure representation. Proc., 45th, Symp. Interface: Com-puting Science & Statistics, Morgantown, WV, Jun 10-13.

[8] Hooft, R. W., Sander, C., & Vriend, G. (1997). Objectively judging the quality of a protein structure from a Ramachandran plot. CABIOS, 13(4), 425–430.

[9] Brevern, A. G. De, Etchebest, C., & Hazout, S. (2000). Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. Proteins, 287:271–287.

[10] Suresh, V., Liu, L., Adjeroh, D., & Zhou, X. (2015). RPI-Pred: predicting ncRNA-protein interaction using sequence and structural information. Nucleic Acids Research, 43(3), 1370–1379.

[11] Shen, J., Zhang, J., Luo, X., Zhu, W., et al. (2007). Predicting protein-protein interactions based only on sequences information. PNAS, 104(11), 4337–4341.

[12] Koh, G., et al. (2012). Analyzing protein-protein interaction networks. Journal of Proteome Research, 11(4), 2014-2031.

[13] Epusz, T., Yu, H., & Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. Nature Methods, 9(5), 471-472.

[14] Reynolds, C., et al. (2009). ProtorP: A protein-protein interaction analysis server. Bioinformatics, 25(3), 413-414.

[15] Wang, Y., Chen, X., Liu, Z.-P., et al. (2012). De novo prediction of RNA-protein interactions from sequence information. Molecular BioSystems, 133–142.

[16] Lu, Q., Ren, S., Lu, M., Zhang, Y., et al. (2013). Computational prediction of associations between long non-coding RNAs and proteins. BMC Genomics, 14(1), 651.

[17] Zhang S, et al. (2016). J. A deep learning framework for modeling structural features of RNA-binding protein targets. Nucleic Acids Research, 44(4), p. e32.

[18] Beal, R., & Adjeroh, D. (2015). Efficient pattern matching for RNA secondary structures. Theoretical Computer Science, 592, 59-71.

[19] Gusfield, D. (1997). Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, New York.

[20] Hofacker, I. L., et al. (2002). Secondary structure prediction for aligned RNA sequences. JMB, 319(5), 1059–66.

[21] Ramachandran GN, et al. (1966). Stereochemical criteria for polypeptide and protein chain conformations. III. Helical and hydrogen-bonded polypeptide chains. Biophys J. 6:849-72.

[22] Ukkonen, E. (1995). On-line construction of suffix trees. Algorithmica, 14(3), 249–260.

[23] Manber, U., & Myers, G. (1993). Suffix arrays: A new method for on-line string searches. SIAM Journal on Computing, 22(5), 935.

[24] Adjeroh, D., Bell, T. C., & Mukherjee, A. (2008). The Burrows-Wheeler Transform: Data Compression, Suffix Arrays, and Pattern Matching, Springer, New York.

[25] Adjeroh D., & Nan, F. (2010). Suffix-Sorting via Shannon-Fano-Elias Codes, Algorithms 2010, 3(2), 145-167.

[26] Lewis, B., Walia, R., et al. (2011). PRIDB: A protein-RNA interface database. NAR, 39(Database), D277-D282.