Formal Grammars and Residual Neural Networks for RNA Secondary Structure Prediction*

Polina Lunina[1,2][0000−0002−7172−2647] ✉, Semyon Grigorev[1,2][0000−0002−7966−0698], and Vadim Abzalov[1,2][0000−0002−0805−0315]

[1] Saint Petersburg State University, 7/9 Universitetskaya nab., St. Petersburg, 199034, Russia
[2] JetBrains Research, Primorskiy prospekt 68-70, Building 1, St. Petersburg 197374, Russia
lunina_polina@mail.ru ✉, semyon.grigorev@jetbrains.com, vadim.i.abzalov@gmail.com

Abstract. RNA secondary structure prediction task is known to be quite critical in computational genomics, therefore, different tools and algorithms are still competing in this field. In this work, we propose to combine formal grammars and neural networks to solve this problem. Our approach utilizes the idea that RNA secondary structure can be viewed as recursive composition of various stems. We describe the most probable types of stems by simple context-free grammar, so that parsing matrix for some sequence would represent all the theoretically possible stems. Then we process these matrices with residual neural networks in order to generate a valid secondary structure. This approach allows to process pseudoknotted structures, non-canonical base pairs and multiple contacts due to flexible nature of neural network training process.

Keywords: CNN · ResNet · Machine Learning · Secondary Structure · Genomic Sequences · Formal Grammars · Parsing.

1 Introduction

Improvement in RNA secondary structure prediction accuracy is one of the key focuses in computational genomics due to its crucial role in functional analysis of RNA molecules. All the diversity of existing secondary structure prediction techniques can be divided into comparative methods that analyse several homologous sequences employing evolutionary approaches [1, 2] and single sequence methods that process one sequence at a time according to some folding constraints, e.g. thermodynamic [3] or statistic [4, 5] rules. One of the challenging parts is pseudoknotted structures processing, because pseudoknots are known to be widely represented in biological data, including functionally important RNA regions,

* Supported by the Russian Science Foundation grant 18-11-00100

nevertheless, building a model that handles them has always been a non-trivial task.

Among other ways, formal grammars can be applied for RNA secondary structure description and some of the algorithms utilize this technique for secondary structure prediction [6, 7]. Due to the probabilistic nature of secondary structure formation laws complicated stochastic (probabilistic) grammars are generally used and the classical way here is to create a grammar that models the whole structure. This approach is known to be quite successful, but it also should be mentioned that building such grammar requires a lot of theoretical and practical difficulties. Therefore, we propose a different way — to encode only stems of secondary structure by simple context-free grammar and leave further processing and probability estimation to machine learning methods that are known to be quite successful in biological data processing [8, 9].

So, in this work, we introduce a new approach for RNA secondary structure prediction which employs the combination of ordinary formal grammars and artificial neural networks. The main ideas were outlined in [10, 11] and this research is conducted to further development of this approach in the context of secondary structure prediction problem. Secondary structure can be formally described as a compositions of stems having different lengths and loop sizes [12], so, we propose to use a simple context-free (not probabilistic) grammar to encode the most common types of stems and search for such stems in the input sequences by matrix-based parsing algorithm. Thereby, the parsing matrix for some sequence will contain the information about whether each subsequence of this sequence can fold to stem or not. This matrix is not yet a representation of a valid secondary structure, because it cannot contain all these stems at once and, besides, there can be more complex elements that are not expressible in terms of our grammar (such as pseudoknots and non-canonical base pairs). Therefore, we propose to process such matrices by neural networks that should filter extra stems and add some missing elements in order to generate a maximal approximation of this sequence secondary structure. So, on the one hand, using neural networks allows to skip full formalization of RNA secondary structure and on the other hand, the grammar provides some sort of basis for neural network training.

2 Proposed Approach Overview

In this section we focus on the theoretical aspects of the proposed approach that underlie all the experimental research presented in the next section. Firstly, we use formal grammars for secondary structure features description and secondly, we apply neural networks for these features processing and solving secondary structure prediction task.

Formal Grammars As it was mentioned before, our approach employs formal grammar not for modeling the whole secondary structure, but for encoding its simple constructional elements — stems.

In figure 1 grammar G_0 that we use in this work as well as in the previous ones is presented. This grammar describes the recursive compositions of stems having height at least 3 (lines **7-12**) and loop size from 1 up to 20 (line **3**). Note that these constants are not mandatory and might be defined experimentally for each task. Also, G_0 allows only conventional base pairs (line **5**) and does not express pseudoknots, because adding the rules for both of these features complicates the grammar unacceptably in the context of performance, therefore, we expect the neural network to handle them instead. Also, we consider only stems of height three or more, because including shorter stems would overload the parsing matrix with too much extra information. So, by this rules, a sequence folds to stem $\iff$ it is derivable from start nonterminal $s1$ of G_0 (line **1**).

```
1  start: stem3<s0>
2  s0: loop | loop stem3<s0> s0
3  loop: nucl*[1..20]
4  nucl: A | U | C | G
5  stem1<s>: A s U | G s C | U s A | C s G
6  stem2<s>: stem1<stem1<s>>
7  stem3<s>:
8        stem1<stem2<s>>
9      | A stem3<s> U
10     | U stem3<s> A
11     | C stem3<s> G
12     | G stem3<s> C
```

Fig. 1: Context-free grammar G_0 for RNA secondary structure stems description

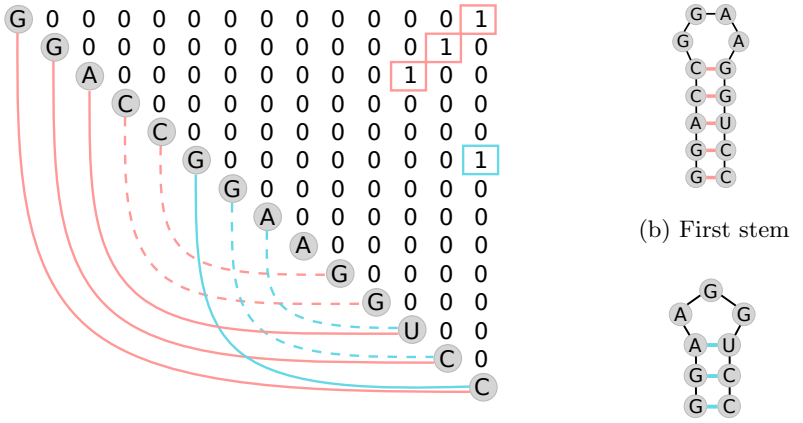
Having a grammar, we want to find all the subsequences of some given sequence that may fold to stems and this is to be done by means of parsing. In all the experiments we use parsing algorithm [13] that is based on matrix operations and demonstrates high performance in practice due to the effective use of GPGPU.

Formally, the result of a matrix-based parsing algorithm for an input string w is an upper-triangular boolean matrix M_P, where $M_P[i,j] = 1 \iff$ the substring $w[i,j]$ is derivable from grammar start nonterminal. From the practical point of view, this means that parsing matrix contains one in a cell $\iff$ a correspondent substring folds to stem according to the rules of a given grammar, so each stem results in a diagonal chain of ones in the matrix, because if sequence $w_1...w_n$ is a stem than it is clear that $w_2...w_{n-1}$ is a stem, $w_3...w_{n-2}$ is a stem and so on while the stem height is at least 3.

In figure 2 we provide the parsing result for a short RNA sequence and show how parsing matrix maps with secondary structure stems. Each one cell describes the stem of height at least 3, so, this sequence contains two subsequences that may fold to stems of the first nesting level. These stems expected hydrogen bonds along with corresponding matrix cells are painted in two different colors.

All nucleotide bonds forming a stem of height three or more are represented by solid lines, moreover, it is obvious that each stem of height three encapsulates stems of heights two and one which are highlighted by dotted lines.

Note that these stems interfere with each other, thereby, secondary structure cannot contain both of them at the same time. So, the parsing matrix for a sequence describes all the theoretically possible folds there, but at the current step we cannot know which one of them would be presented in the real secondary structure. Moreover, G_0 has certain limitations, such as stem height, loop size and possible base pairs, so, some of the required stems may be missing in the parsing matrix. While creating a grammar we were guided by two competing ideas: to cover as many types of stems as possible and to stay with adequate amount of extra information in parsing matrix along with acceptable time costs for parsing.



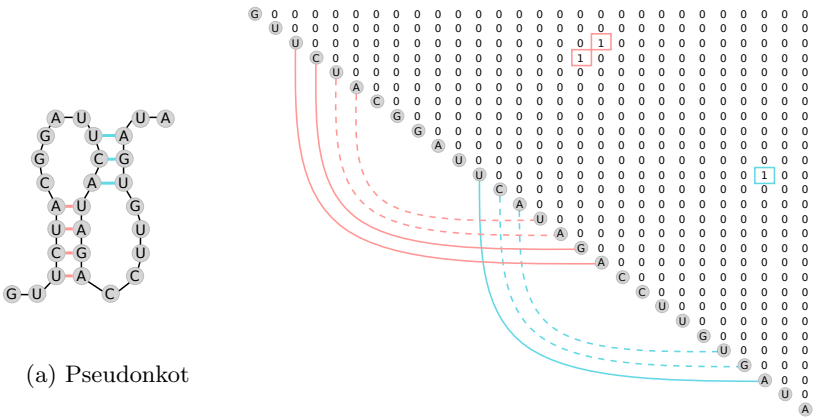
(a) Parsing matrix

(b) First stem

(c) Second stem

Fig. 2: Stems extracted from RNA sequence

Let us consider pseudoknotted structures processing in the context of the proposed approach. Even though it is clear that pseudoknot is not explicitly expressed in G_0, it consists of two stem-loop structures having half of one stem located between two halves of another, so, each of these stems separately can be derived by the rules of G_0. Therefore, pseudoknots will be reflected in the parsing matrix and handling them becomes an additional task for neural network. In figure 3 an example of simple pseudoknot along with corresponding parsing result is provided. Two stems of this pseudoknot are highlighted with two different colors and it can be seen that all the related nucleotide bonds are presented in the parsing matrix, although at this point it is not determined whether sequence contains a pseudoknot or it just has two possible folds in terms of grammar.



(a) Pseudonkot

(b) Parsing matrix

Fig. 3: Pseudoknots processing in terms of G_0

Neural Networks The final step of the proposed solution is to process parsing matrices by a neural network in order to achieve a maximal similarity with the expected sequence secondary structure. Therefore, we need to specify data preparing pipeline and to develop an optimal architecture for a problem at hand.

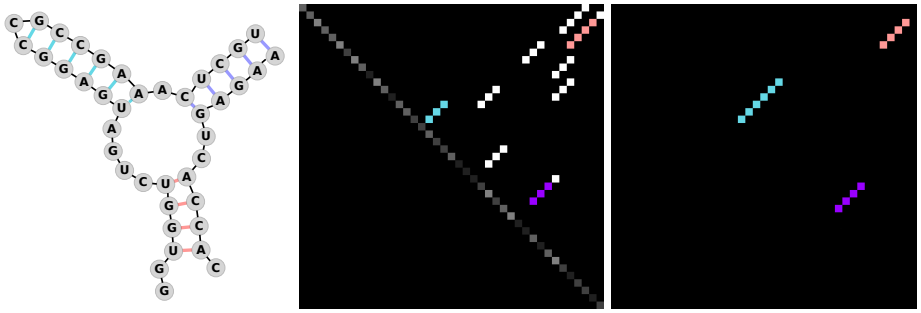
Data The input data for neural network (parsing matrices) was described in the previous section and now let us define the reference data source and format. There are specialized biological databases containing RNA chains of various organisms along with their secondary structures obtained by reliable methods and such data is known to be the best for algorithms training and validation. These databases may store data in different formats (dot-bracket, connectivity table and others), so, we need to choose convenient for our experiments and compatible with others format.

One of the ways of RNA secondary structure formal representation is so-called contact map, which for an input string w is a boolean matrix M_C, where $M_C[i,j] = 1 \iff$ nucleotides in positions i and j form a hydrogen bond (or, to put it simply, a contact) in secondary structure. Consider the discussed earlier parsing matrix for the same sequence w that has 1 in the cell $[i,j] \iff$ subsequence $w[i,j]$ folds to a stem. It is clear that the first and the last nucleotides of every stem form a contact, therefore, we can easily transfer between parsing matrix and contact map definitions and view the parsing matrix as a sort of a contact map, so, this format is acceptable for our experiments. Note that if parser finds a stem of height three than we will see only one cell with 1 in matrix, but such stem always wraps a stem of height two which wraps a stem of height one, so, we are always missing two contacts, therefore, after pars-

ing we should set $M_P[i-1, j+1] := 1$, $M_P[i-2, j+2] := 1$ if $M_P[i][j] = 1$, $i = 0..size(M_p), j = i{+}1..size(M_p)$ for complete equality of these two structures.

So, neural network should take parsing matrices as inputs and contact maps as desired outputs for the same set of RNA sequences. For convenience, we transform both matrices to black-and-white images by replacing zero cells with black pixels and one cells with white pixels. Also, we code RNA sequence at the input image main diagonal by four types of gray pixels corresponding to the four possible nucleotides in case the chain itself contains any important information about secondary structure formation patterns.

In figure 4 we provide two-dimensional secondary structure visualization for RNA sequence along with neural network input and reference images that were made from parsing and contact matrices respectively. Contacts belonging to the three stems presented in this sequence are highlighted with three different colors in all pictures (so, input and reference images are actually grayscale, colored pixels are only for clarification). It can be seen that not all of the stems found by parser are presented in the real secondary structure. Moreover, the parsing result is missing several contacts due to the fact that they were formed by non-canonical nucleotide pairs $A - G$ that are not expressed by grammar G_0.



(a) Secondary structure visualization

(b) Input image for neural network

(c) Reference image for neural network

Fig. 4: The correspondence between secondary structure stems and pixels in input and reference images

Parallel ResNet One of the popular architectures for complicated image processing tasks is residual neural network based on adding skip connections between blocks of layers [14]. ResNets solve the problem of vanishing gradient and allow to effectively use deep convolutional networks.

In this work, we developed a new architecture that showed its applicability for secondary structure prediction task during experimental research. The idea is to build n identical residual networks that are trained independently on the same data, connect their n outputs with weighted sum and hang this result over

to the final residual unit that directly generates output. This parallel residual architecture along with the scheme of a typical residual unit is presented in figure 5 and in this work we set $k := 10$, $n := 4$ based on empirical evidence. We believe that the advantage of this parallel technique is that these separate networks are able to find different types of features in data and some sort of voting system allows the whole model to decide for the particular pixel whether each network behaves correctly or not.

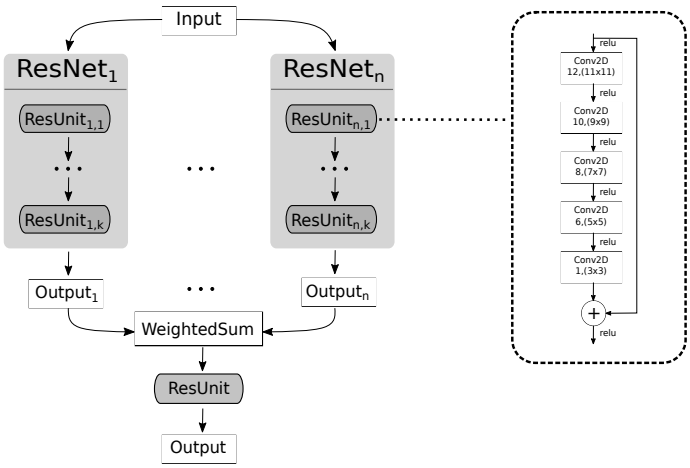


Fig. 5: Parallel residual neural network architecture

2.1 Experiments

References

1. I. L. Hofacker and P. F. Stadler, "Automatic detection of conserved base pairing patterns in rna virus genomes," *Computers & chemistry*, vol. 23, no. 3-4, pp. 401–414, 1999.
2. F. Tahi, M. Gouy, and M. Régnier, "Automatic rna secondary structure prediction with a comparative approach," *Computers & chemistry*, vol. 26, no. 5, pp. 521–530, 2002.
3. M. Hamada, H. Kiryu, K. Sato, T. Mituyama, and K. Asai, "Prediction of rna secondary structure using generalized centroid estimators," *Bioinformatics*, vol. 25, no. 4, pp. 465–473, 2009.
4. S. R. Eddy and R. Durbin, "Rna sequence analysis using covariance models," *Nucleic acids research*, vol. 22, no. 11, pp. 2079–2088, 1994.

5. C. B. Do, D. A. Woods, and S. Batzoglou, "Contrafold: Rna secondary structure prediction without physics-based models," *Bioinformatics*, vol. 22, no. 14, pp. e90–e98, 2006.
6. B. Knudsen and J. Hein, "Pfold: Rna secondary structure prediction using stochastic context-free grammars," *Nucleic acids research*, vol. 31, no. 13, pp. 3423–3428, 2003.
7. M. E. Nebel and A. Scheid, "Evaluation of a sophisticated scfg design for rna secondary structure prediction," *Theory in Biosciences*, vol. 130, no. 4, pp. 313–336, 2011.
8. S. Higashi, M. Hungria, and M. Brunetto, "Bacteria classification based on 16s ribosomal gene using artificial neural networks," in *Proceedings of the 8th WSEAS International Conference on Computational intelligence, man-machine systems and cybernetics*, pp. 86–91, 2009.
9. D. Sherman, "Humidor: Microbial community classification of the 16s gene by training cigar strings with convolutional neural networks," 2017.
10. S. Grigorev. and P. Lunina., "The composition of dense neural networks and formal grammars for secondary structure analysis," in *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 3 BIOINFORMATICS: BIOINFORMATICS,*, pp. 234–241, INSTICC, SciTePress, 2019.
11. P. Lunina and S. Grigorev, "On secondary structure analysis by using formal grammars and artificial neural networks," in *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pp. 193–203, Springer, 2019.
12. M. Quadrini., E. Merelli., and R. Piergallini., "Loop grammars to identify rna structural patterns," in *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 3 BIOINFORMATICS: BIOINFORMATICS,*, pp. 302–309, INSTICC, SciTePress, 2019.
13. R. Azimov and S. Grigorev, "Context-free path querying by matrix multiplication," in *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA '18, (New York, NY, USA), Association for Computing Machinery, 2018.
14. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.