Single RNA Secondary Structure Prediction based Dynamical programming algorithms: to parallelize or not?

Bisera Chauleva[1]

[1] Faculty of Computer Science and Engineering at
UIST "St.Paul the Apostle",
Ohrid, Republic of North Macedonia
bisera.cauleva@cse.uist.edu.mk

Ljubinka Sandjakoska[1]

[1] Faculty of Computer Science and Engineering at
UIST "St.Paul the Apostle",
Ohrid, Republic of North Macedonia
ljubinka.gjergjeska@uist.edu.mk

Atanas Hristov[3]

[3] Faculty of Information and Comunication Sciences at
UIST "St.Paul the Apostle",
Ohrid, Republic of North Macedonia
atanas.hristov@uist.edu.mk

***Abstract*— RNA Secondary Structure Prediction has a huge importance for Bioinformatics. Over the last decade, Dynamical Algorithms used for that purpose reached performance bottlenecks, with data produced by RNA sequences. The main idea of this paper is to answer the question – to parallelize or not? We aim to achieve a better performance over different algorithms. The chronological development of algorithms is followed and we try to obtain better execution time accordingly, where we introduce comparison between serial and parallel version of the algorithm. As a performance measurements are obtained the Time Complexity and Accuracy Level with accent on the best algorithm for the purpose needed.**

Keywords—Bioinformatics, Dynamical Programming, RNA Secondary Structure Prediction, Parallelization,

I. Introduction

RNA is defined as the second most important element after DNA, which is complex molecule that takes function in cellular protein synthesis. Its structure is used in encoding and decoding genes, as well as regulation of their expression in living organisms. RNA is constructed of ribose nucleotides connected with phosphodiester bonds, forming strands of varying lengths. The nitrogenous bases in RNA are adenine, guanine, cytosine, and uracil. The three-dimensional structure of RNA is critical for its stability and function, since bases could be modified in different ways by cellular enzymes and manipulation of groups to the chain. RNA could be defined as a structure which features are between linear molecule and 3-D structure. If secondary structure is taken in consideration, the RNA is composed of double stranded regions, when the single linear RNA is folded upon itself. Different structures are evaluated further with some algorithms such as: *Stem Loops or Hairpins*; *Bulge Loops*; *Interior Loops*; *Junctions or Multiloops*; *PseudoKnots* etc.RNA secondary structure was discovered by X-Ray methods, which were extremely hard and expensive to be performed for all possible RNA sequences. Therefore a new concept for faster and efficient formation of a secondary structure was invented, known as computational prediction of RNA secondary structure. One type of computational prediction is by dynamical programming (DP) which is a useful technique for complex problems like the RNA structure. With the help of DP algorithms calculation is done over one major problem when subdivided on multiple smaller problems, therefore efficient prediction of the RNA structure could be performed. There are different algorithms using this method that are going to be presented in our work, such as:

- *Nussinov-Jacobson Algorithm* – for formation of secondary structure of RNA based on folding upon itself. That base-pairs can form secondary structure. This algorithm is known as the first algorithm for that purpose, with time complexity of O(n^3) [1]. Different type of implementation of this algorithm is known as Four Russians Algorithm with perfect time complexity of O(n^3/ log(n))[4];

- *Minimal-free Energy Algorithm* (Zucker's Algorithm) where the main focus is on amount of free energy expressed by each adjacent base pair. Since different RNA structure has different amount of free energy presented, this algorithm gives great accuracy for shorted RNA sequences, with time complexity of O(n^4) [3];

- *Maximum Expected Accuracy* (MEA) - mostly focused on partition function calculation based on McCaskill's Algorithm, that utilizes the free energy change with the usage of nearest-neighbor parameters. MEA predict base pair probabilities as well as probabilities of nucleotides being single-stranded, with time complexity of O(n^3) [6] [7];

- *Pseudoknotted Algorithm* – simple dynamic programming algorithm for RNA secondary structure prediction with pseudoknots which is mainly based on the theoretical approach of Akutsu's algorithm, with time complexity of O(n^5)[10][11];

In this work, all of the mentioned algorithms are going to be evaluated. The evaluation is followed by parallelization with OpenMP in order to find proved answer of the question – *to parallelize or not* where we are interested in better performance for different sequence lengths. The rest of the paper is organized as follow. In the second section theoretical background of the RNA second structure prediction algorithms is given. In the third, the implementation and testing of parallelization with OpenMP is presented, followed by performance and accuracy discussion sections. Before the list of used references, in the last section, the experimental setup that include the testing platform and the description of the requirements is given.

II. RNA Secondary Structure Predicting Algorithms

A. *Nussinov-Jacobson Algorithm*

Nussinov-Jacobson algorithm is proposed by Nussinov and Jacobson in 1978 [1]. It is defined as algorithm for secondary RNA structure prediction based on the folding principle, when the RNA strand is folding onto itself, without

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

taking in consideration formations like pseudoknots. Mainly it considers usage of the maximum amount of base pairs for optimization of the score. It is based on the usage of the standard 2D array. As a score values of X_i and X_j are used in form of a matrix M[i][j]. This algorithm, take in considerations two cases: if leftmost base is unpaired or paired with other base. We use three stages when constructing the Nussinov-Jacobson algorithm [1]:

1.Initialization step, in this step the scoring of matching elements present on the main diagonal and the diagonal below it are done, where the rules respected are: M[i][j] = 0 for i=1 to L, and M[i][i-1] = 0 for i=2 to L, where L is the length of the RNA

2.Recursive step, which considers fulfilment of the matrix, using the four major conditions like: if *i* is unpaired, added onto a structure for *i + 1, j*, for it the matrix will follow M[*i*+1][*j*] (ith residue is hanging off), if *j* is unpaired, added onto a structure for *i. j* – 1, where the matrix follows M[*i*][*j j*-1] (*j*-th residue is hanging off), if *i* and *j* base pair are added on to a structure for formation of *i*+1, *j*-1, where the matrix has M[*i*+1][*j j*-1] + S (x_i, x_j) (*i*-th and *j*-th residues are paired and if x_i is complement of x_j, then S(x_i, x_j) = 1; otherwise it is 0, and finally if *i* and *j* are making a base pair but not to each other, the structure for *i..j* adds together to a substructures, for two sub-sequences, *i..k* and *k + 1..j* (making a bifurcation). The matrix will follow *M[i][j] = MAX i<k<j (M[i][k] + M[k+1][j])* (merging two substructures). (Fig 1.)

3.Traceback step, considers the formation of actual secondary structure of RNA sequence based on the trace-back from the given scores in the matrix, which are filled in by the previous steps. This step is missed in the algorithm used in this project, it can be formed by the scores obtained from the matrix.

In order to implement these steps practically must use computation of M matrix by diagonals and within a diagonal from top to bottom. Calculated run time is O(n^3). Even if the algorithm is written using two-dimensional array notation for M, we need only the upper triangle of M. Therefore if we want to have memory efficient implementation, with the usage of either mapping of the upper triangle into a 1D array or dynamically allocated 2D array with variable size rows, must be used. In both cases it is expected to have memory for n*(n+1)/2 elements of M [2].

For better time complexity we consider another type of implementation known as Four Russian Algorithm [4]. The contribution is both theoretical and practical, since the basic RNA-folding problem is often solved multiple times in the inner-loop of more complex algorithms and for long RNA.

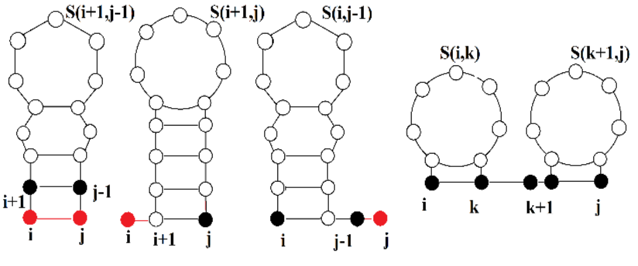


Fig. 1. *Recursive step possible cases with pairing*

As RNA input for this folding problem we take a string K of length n as representative of nucleotides, and an element d to show the maximum distance between two sites of a match. In order to obtain matching we use a set M which holds the pairs that are disjoints to a set K which hold other sides. If pair (*i, j*) is in M, then the nucleotide *i* will match to the one on site of *j*. We obtain a permitted match if the nucleotides at sites *i* and *j* are complimentary, and |*i* − *j*| > d. M is non-crossing or nested if and only if it does not contain any four sites *i* < *i`* < *j* < *j`* where (*i, j*) and (*i`, j`*) are matched in M. If we place the sites of K in a circular order, and draw a straight line between them, then in each pair in M, will have a non-crossing pair if and only if no two straight lines cross. Finally, a permitted matching M is a matching that is non-crossing, where each match in M is a permitted match [4]. From the following we obtain a cubic-time algorithm is we consider the work over three nested loops, for *j, i* and *k* that make increments of O(n) times when entered. The speed-up can be obtained where instead of incrementing k through each value from *j − 1* down to *i + 1*, it is practical to make a combination into groups of size q, which gives constant amount of time per group. This modification is done with the introduction of a vector Vg. [4]

B. *Minimum Free Energy (Zuker's) Algorithm*

In order to calculate the minimum energy we are adding experimentally predetermined values for each base pair, which is found in the dynamic programming matrix. The free energy depends on the sequence part of actual segment and the most adjacent base pairs. The total free energy is the sum of all increments. This concept is implemented as algorithm for RNA secondary structure prediction, available as MFold. The approach is also known as Minimum Free Energy (MFE) and was developed by M. Zuker [5]. There are certain limitations to MFE method such as that within the method, energies of bulge loops and single non-canonical pairs are not predicted. Zuker's Algorithm [9] mainly uses approach that divides a secondary structure, such that loops also known as graphs are used, and the free energy value is given based upon those graphs. Calculation of the lowest free energy structure, gives us the optimal structure of RNA molecule with consideration of maximum base pair amount. Zuker's algorithm takes in account different energies for calculation.

Zuker' s algorithm defines two matrices *W(i,j)* and *V(i,j)*, where *W(i,j)* is the total free energy of subsequence *i* to *j*. For the *V(i,j)* is defined as the total free energy of subsequence i to *j* if *i* and *j* pairs, otherwise, *V(i,j)* = ∞ consecutively *FH(i,j)* is the energy of hairpin loop *i…j*. Whereas *FL(i,j,h,k)* is the energy of 2nd order loop such as stack region, bulge loop and interior loop *i…h…k…j*. The last item is the energy for bifurcation loop, where item repeats over *i+1<k < j-1* because *i* and *j* must be a base pair, otherwise *V(i,j)* = ∞. From where *W(1,L)* gives the final total minimum free energy (Fig. 2).

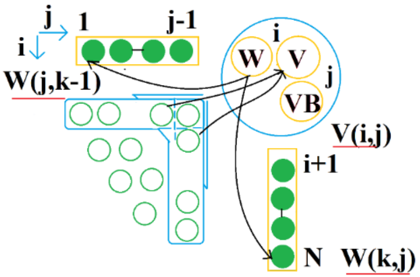


Fig. 2. Internal Loops Wand V matrix calculation depicted

C. Maximum Expected Accuracy (MEA) Algorithm

The concept of maximum expected accuracy algorithm, is a concept where in order to predict a secondary structure of RNA we need to use the technique of partition function calculation [10], with which we predict base-pair probabilities. These probabilities are then used by dynamic programming algorithm, which we can find in the RNA structure online service where the native *C* code for development and further improvement could be obtained.

MFE method finds one specific best guess for secondary structure of RNA, due to that it has drawbacks such as, when we use only one conformation at equilibrium instead of more. Compared when using MEA, assumed conformation from more conformations has probability of base pair based on a partitioning McCaskill's function algorithm. With this method usually high probability base pairs are chosen, and they have higher accuracy level. Additionally the base pair probability is less prone to change than it is in the thermodynamic measures of the MFE, which proves that some errors are overcome. MEA method uses derivation temperature of 37°C which is optimal, even if we make testing over RNA sequences which are from different organisms functioning on different temperatures. There are certain methods that have modification over this parameter too. Another characteristic of this method is the prediction of MEA structure to use only independent base pairs, which is not right when considering structures like helix one, where cooperation between pairs is present. Still there are methods for advancing this basis of algorithm that uses this characteristics to obtain higher accuracy level.

Constructing blocks of MEA method are:

a) Nearest Neighbor Parameters(NNP), which are the one used by Watson- Crick helices that go under the partitioning function developed by McCaskill;

b) McCaskill's partitioning function, which was developed by McCaskill in 1990 [10], and is mainly based on original statistical problem where the calculation of the partition function must be made first, in order to obtain further specific quantities of thermodynamic interest. The most useful part of the equilibrium ensemble of structures is in the binding probabilities between base pairs. Here the notice is made on the probabilities which are not locally determined by the sequence, but instead each probability has some effect over the equilibrium sum of the structures. From here, the formed matrix gives information about the global ensemble of structures in equilibrium. We need to establish that the equilibrium ensemble made of summed probabilities from bounded bases propose a direct comparison to enzymatic and chemical modification experiments, with goal to detect any modifications in bases exposition.

McCaskill's partition function algorithm is composed of two parts. First partitioning function scores are obtained and after that the probabilities of base pairs are calculated. Also known as folding and backtracking steps. The folding part is corresponding to the Zuker's algorithm, whereas the backtracking is completely authentic.

QB_{ij}, is defined as the partition function of the substrings *i* and *j* which are paired and Q_{ij} is taken for the unconstrained partition function. From there the partition function of whole molecule would be given as Q = Q1n. If we end up with *i* and j which are paired, then we can form a hairpin loop or interior loop *i**j or h*l or eventually a loop with multiple components.

The variables Q_m and Q_{m1} are used for handling of the multiloop formations. Also we have variable of Qa which is used for the size of the internal loops which makes modification in the time from $O(n^4)$ to $O(n^3)$. In the backtracking of the algorithm for the pairing probabilities P_{ij} we get value from the partitioning function QB_{ij} and Q_{ij} mentioned before.

For the implementation purposes we use the simplified version of the Nussinov-like energy scoring scheme, where each pair formed in a structure has a contribution to a fixed energy term Ebp which is aside of its context. From here we form two dynamic programming tables Q and Qbp. The partition function for a sub-sequence from position *i* to position *j* is provided by Q_{ij} . Array QBp holds the partition function of the sub-sequences, which form a base pair or 0 if base pairing is not possible.

Recursive functions are used to compute Q and QBp. The input data are RNA sequence S as a chain of nucleotides. We have specification of minimal loop length l (also defined as minimal number of enclosed positions), energy weight of base pair Ebp and normalized temperature RT. The memory complexity of the arrays is $O(n^2)$, while the time complexity of a direct implementation of this algorithm is $O(n^3)$ in the sequence of length n.

D. Pseudoknotted Algorithm

When choosing which algorithm we can use it is difficult to choose upon high amount of algorithms suggested because some algorithms lack the accuracy of prediction in the pseudoknots considering the following characteristics: based on lack of knowledge in area of energy models we have more difficulty in discovery of secondary structures, the folding principle in formation of a structure can be affected by kinetic energies, ligand-binding, interactions in transition, and finally small amount of experiments result in small knowledge in pseudoknot formation, therefore only H-type of pseudoknots are found with most of the algorithms.

For the suggested dynamic programming algorithm in first step we use the algorithm for prediction of MFE structure which will have up to 100 suboptimal structures predicted. Additionally this algorithm will be able to generate dot plot which will provide nucleotides i and j, with MFE structure containing i-j base pair. The ΔG° values will be calculated using the current Turner nearest-neighbor parameters but with the multi-branch loop [13]. Pseudoknot helix list, H, follows with corresponding helix energies. There is some criteria that needs to be pleased to get into helix list, such as sequence size restriction to be longer than 100 nucleotides, due to which examined sequences are of 200, 500 and 1000 nucleotides, also ΔG° must be 25% of the free energy of MFE. The ΔG° of Hi will be obtained from the nearest-neighbor AU/GU pairs.

Filtration of helices H is done in some particular steps. A helix Hi is accepted into H if it has more than 3 base pairs. Helices are going through comparison with the MFE structure. If they have more than 50% base pairs paired in the MFE they are discarded. For each Hi, a new set of structures, taken lowest of all MFE structures and up to 99 suboptimal structures, is generated by the dynamic programming algorithm, with Hi prohibited from pairing. [11]

Consecutively, base pairs from Hi are restored to the structures. The ΔG° of each structure is incremented by the free energy of the corresponding helix Hi. All unique structures are added to S. In S, an entropic cost of the pseudoknot formation is generated by the ΔG°PK. In order to be a pseudoknot it must have at least two helices in a formation, one side of a helix to match with base pairs in the second helix. (Fig.3) Formed structures in pseudoknot can be: structure (SS) with single-stranded nucleotides inside the pseudoknot and (NE) representing number of nested helices inside the pseudoknot, also the IL (N) defined as the amount of in-line helixes of length N. [12]

Before the intervening structures are calculated, the pseudoknot are in advance calculated with filling single and grouped mismatches with base pairs and removing isolated pairs.[15] Helices containing a single bulged nucleotide are counted as a single helix. Terms e and f give the values of entropic penalty by the distance between carbons of neighboring unpaired nucleotides and across a single base pair. In-line helix frequencies P1 and P2 are constant energy parameters that include Boltzmann constants and temperature terms and must be determined empirically. ΔG°PK is added to the total ΔG° of each pseudoknot-containing structure. (Fig. 3) [15]

The sorting of S is done based on the total energy that each structure has. As the rule says, the first 20 that have lowest MFE are discarded. The one which will be executed are put in the Window parameter which separates them from other structures. In order to be part of this, the structure must have equal amount or more than the window base pairs amount. Usually a default value for the window parameter is given based on a length of a sequence. Finally, structures that are higher in folding MFE are discarded with the help of a parameter which accounts the maximum percentage of energy difference. By default the value is 10%.

Coaxial stacking of helices stabilizes pseudoknot formation and is included indirectly in the energy function. [11] Coaxial stacking has effect over the helices which are chose in order to assemble the pseudoknots.

III. Parallelization with OpenMP

A. Implementation

For the purpose of implementation we are going to evaluate each algorithm with corresponding paralleization segment of code considering OpenMP rules [18].

- For the Nussinov-Jacobson Algorithm, we have considered paralleization over the main recursive segment where the scoring matrix over diagonals and fullfilment of it is made;
- For the Zuker's MFE Algorithm the practise would be the same, since we have one starting recursive function taken for initialization and calculations over two matrices and different structures;
- MEA Algorithm taken advantage of the partitioning McCaskill's function, therefore the parallelization would be perforomed over it;

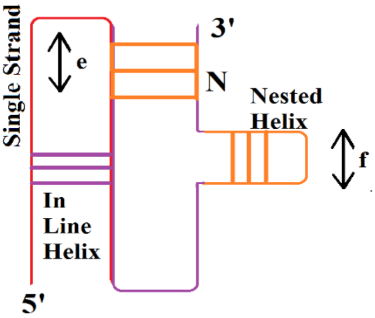


Fig. 3. Formation of pseudoknots with given penalties for in-line helix, single stranded helix and nested helix

- Finally, for the Pseudoknotted Algorithm we can take advantage of the structure S. The structures need to go under sorting based on total energy which takes a lot of time and takes advantage of parallelization.

B. Testing and Results

We are going to provide results from testing over serial implementation of algorithms.

For testing purpose we are using HSBGPG Human gene for bone gla-protein (BGP) in FASTA form. [20]

The time complexities of the represented algorithms are: Nusssinov-Jacoboson Algorithm defines as O(n^3), Nussinov's Four Russian Algorithm defined as O(n^3/log(n)), Zuker's MFE and the time complexity of O(n^4), MEA algorithm with the time complexity of O(n^3), the Pseudoknotted algorithm with the highest time complexity of O(n^5). Obtained results and comparison are given in the *Fig. 4*). From the provided results we can make comparison and obtain number of times we have obtained speed up. We can test the speed up with the help of Amdahal's Law, formula given (1).

$$Speedup = \frac{1}{(1-p)+\frac{p}{N}} \tag{1}$$

According to the given formula obtained results of speed up are given on table (*Fig 5*). From calculated speedup we can make conclusion that parallelization can decrease execution time for at least 2.67 times.

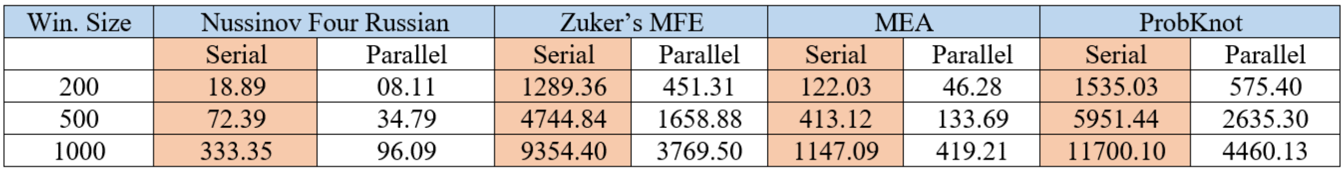
IV. Performance and Accuracy

For the purpose of accuracy level comparison of the algorithm we use the classical benchmark specification known as PPV or positive-predictive value, which is founded on the base-pair prediction accuracy. How sensitive is this benchmark value is given with the percentage of obtained base pairs that are correct, also the PPV value can be defined as a value of a structure that provides the amount of predicted pair.

These characteristics are calculated with the following formula (2) and (3).

$$PPV = \frac{Number\ of\ Correct\ Predicted\ pairs}{Total\ Number\ of\ Predicted\ Base\ pairs} \tag{2}$$

$$Sensitivity = \frac{Number\ of\ Correct\ Predicted\ pairs}{Total\ Number\ of\ known\ Pairs} \tag{3}$$



Win. Size	Nussinov Four Russian		Zuker's MFE		MEA		ProbKnot	
	Serial	Parallel	Serial	Parallel	Serial	Parallel	Serial	Parallel
200	18.89	08.11	1289.36	451.31	122.03	46.28	1535.03	575.40
500	72.39	34.79	4744.84	1658.88	413.12	133.69	5951.44	2635.30
1000	333.35	96.09	9354.40	3769.50	1147.09	419.21	11700.10	4460.13

Fig. 4. Execution Time of proposed Algorithms tested for Serial and Parallel Implementation of Code over different Window Size

After calculation performed over higher training set, conclusion made are based on average results for the sensitivity of algorithms and PPV:

a) Nussinov-Jacobson's Algorithm has Sensitivity average score of 0.65 or 65% and PPV of 0,48 or 48%

b) Zuker's (MFE) Algorithm has Sensitivity average score of 0.73 or 73% and PPV of 0.66 or 66%

c) MEA Algorithm has Sensitivity average score of 0.72 or 72% and PPV of 0.67 or 67%

d) Pseudoknotted Algorithm has Sensitivity average score of 0.72 or 72% and PPV of 0.76 or 76%

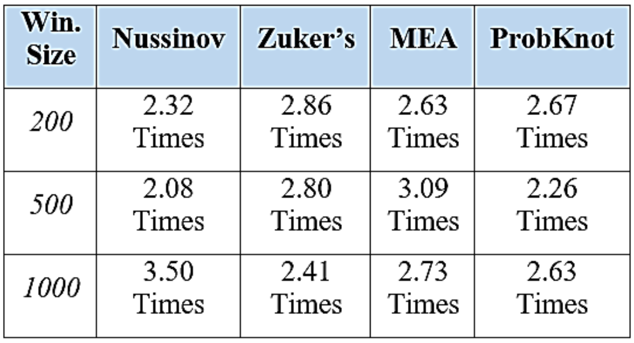
From where we can conclude that the smallest sensitivity level and PPV has the Nussinov algorithm as oldest method, and the best results has the Pseudoknotted algorithm as expected since it is the newest and predicts all kinds of structures.

V. Discussions and Conclusion

From the parallelization of introduced secondary structure predicting RNA algorithms, a couple of conclusions can be made. First discussion of the most important algorithms in chronological order of occurring in dynamical programming branch was given.

After the serial implementation and introduction of parallelized version, comparison in the time complexity and accuracy level was performed, from where a couple of assumptions were made.

First of all, the fastest algorithm of all was the Four Russian's algorithm or so known as the new generation of the Nussinov's algorithm, but not always the fastest algorithm means the most accurate algorithm. Considering the PPV and Sensitivity tests based on benchmarks, calculations show that the most accurate of all is the pseudoknotted algorithm, as the most advanced which is in capability to predict any kind of structure. But as a cost for this advantage, this algorithm is the slowest.



Win. Size	Nussinov	Zuker's	MEA	ProbKnot
200	2.32 Times	2.86 Times	2.63 Times	2.67 Times
500	2.08 Times	2.80 Times	3.09 Times	2.26 Times
1000	3.50 Times	2.41 Times	2.73 Times	2.63 Times

Fig. 5. SpeedUp for each tested Algorithm calculated according Amdahal's Law Formula

Depend on that how prediction of secondary structure should be done from sequence of RNA, due to gene expression, encoding and decoding of genes, it could be chosen across lots of different approaches.

In the branch of dynamical programming in bioinformatics, from the most popular and stable were chosen Zuker's as the basis for all other newly created algorithms, until reaching the last one which predicts all kinds of secondary structure formations, such as the pseudoknots. According to that, as the time goes by, the algorithms become more complex but also more capable of predicting any kind of RNA secondary structure formation from a given sequence, which again gives the chance for lots of advancements in the bioinformatics field.

In order to decrease the time complexity, we use the parallelization option with OpenMP, which showed a high amount of improvement in the execution time for about 3 times per each algorithm that has huge importance when working with longer sequences.

VI. Testing Platform and Requirements

For the purpose of testing of the introduced dynamical algorithms was used the platform CPU Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, where were tested serial and parallelized version of the algorithms.

For the algorithms: Nussinov-Jacobson, Zuker's and Four Russian implementation of Nussinov, were used publically available source codes with elementary changes implied.

For the algorithms MEA and pseudoknotted prediction were used source codes taken from the RNA structure developers package where small improvements were introduced and tested on our platform. Source code in all algorithms tested was C/C++ develop and tested in the Visual Studio 2017 Package with additional support of OpenMP for the purpose of parallelization.

Characteristics of building the RNA structure algorithms MEA with McCaskill's function and pseudoknotted algorithm are the following:

a) C++ class libraries encapsulated in the I/O functions of RNA structure and also the secondary structure prediction and analysis methods were used. The classes are designed to be easily included in C++ projects;

b) Text interfaces;

c) Thermodynamic parameters, for nearest neighbor parameters used for the purpose of prediction of the stability in secondary structures. Here we include the change of free energy parameters at 37°C and the change of enthalpy parameters which are used for the conformational stability and the arbitration of temperature. These are taken from the Turner's group.

Finally in order to access the code and build we need Windows GUI: Microsoft Foundation Classes (MFC) as found in Microsoft Visual Studio 2005 or later and the Intel C++ compiler.

For the purpose of parallelization of the code, we need to enable the OpenMP option in the Visual Studio package.

As RNA input sequence we use the FASTA file format, saved as .txt outside the algorithm. Input module is defined as one which takes the FASTA file and extracts the sequence into a string.

References

[1] Palkowski, M., & Bielecki, W. (2017). Parallel tiled Nussinov RNA folding loop nest generated using both dependence graph transitive closure and loop skewing. *BMC bioinformatics*, *18*(1), 1-10.

[2] Zhao, C., & Sahni, S. (2017). Cache and energy efficient algorithms for nussinov's rna folding. *BMC bioinformatics*, *18*(15), 518.

[3] Zhao, C., & Sahni, S. (2017, October). Efficient RNA folding using Zuker's method. In *2017 IEEE 7th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)* (pp. 1-6). IEEE.

[4] Venkatachalam, B., Gusfield, D. & Frid, Y. "Faster algorithms for RNA-folding using the Four-Russians method. Algorithms"2014, Mol Biol, 2014, pp.9-5.

[5] Will, S., Jabbari, H. "Sparse RNA folding revisited: space-efficient minimum free energy structure prediction. Algorithms" 2016,Mol Biol,2016, pp.11-7.

[6] Clote, P., Lou, F. & Lorenz, W.A. "Maximum expected accuracy structural neighbors of an RNA secondary structure." 2017 BMC Bioinformatics,2017, pp.13- S6

[7] Aghaeepour, N., Hoos, H.H. "Ensemble-based prediction of RNA secondary structures." 2013 BMC Bioinformatics, pp.14-139.

[8] Zhang, B., Yehdego, D.T., Johnson, K.L. et al. "Enhancement of accuracy and efficiency for RNA secondary structure prediction by sequence segmentation and MapReduce."2013 BMC Struct Biol,2013, pp.13-S3

[9] Wu, Y., Shi, B., Ding, X., Liu, T., Hu, X., Yip, K. Y., ... & Lu, Z. J. (2015). Improved prediction of RNA secondary structure by integrating the free energy model with restraints derived from experimental probing data. *Nucleic acids research*, *43*(15), 7247-7259.

[10] Palkowski, M., & Bielecki, W. (2019, September). Parallel cache-efficient code for computing the McCaskill partition functions. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 207-210). IEEE.

[11] Sharma, D., Singh, S., & Chand, T. (2015, January). RNA Pseudoknot: Topology and prediction. In *2015 International Conference on Computer and Computational Sciences (ICCCS)* (pp. 244-248). IEEE.

[12] Song, Y., Liu, C., & Li, Y. (2015). A new parameterized algorithm for predicting the secondary structure of rna sequences including pseudoknots. *The Computer Journal*, *58*(11), 3114-3125.

[13] Liu, Z., Zhu, D., & Dai, Q. (2015, December). Predicting Algorithm of RNA Folding Structure with Pseudoknots. In *2015 11th International Conference on Computational Intelligence and Security (CIS)* (pp. 34-37). IEEE.

[14] Liu, Z., Kong, Q., Fu, Y., Ye, H., Zhao, S., Su, X., ... & Wang, Y. (2017, December). The Algorithm and Scheme of Prediction in RNA Folding Structure with Pseudoknots. In *2017 13th International Conference on Computational Intelligence and Security (CIS)* (pp. 469-474). IEEE.

[15] Liu, Z., Liu, F., Kong, Q., Hao, F., & Zhao, H. (2018, November). Algorithm and Scheme in RNA Structure Prediction Including Pseudoknots. In *2018 14th International Conference on Computational Intelligence and Security (CIS)* (pp. 196-200). IEEE.

[16] Stern, H.A., Mathews, D.H, "Accelerating calculations of RNA secondary structure partition functions using GPUs,",2013 Algorithms Mol Biol 2013, pp.8-.29.

[17] Zhang, H., Zhang, C., Li, Z., Li, C., Wei, X., Zhang, B., & Liu, Y. (2019). A new method of RNA secondary structure prediction based on convolutional neural network and dynamic programming. *Frontiers in genetics*, *10*.

[18] OpenMP API C/C++ Syntax Reference Guide, "OpenMP application program interface version 4.5,"2015.[Online].Available: https://www.openmp.org/wp-content/uploads/OpenMP-4.5

[19] Microsoft Visual Studio Software for editing and code Evaluation, https://visualstudio.microsoft.com/vs/whatsnew/

[20] Human gene for bone protein – Nucleotide - Retrieved on 20.12.2019 from http://www.cbs.dtu.dk/services/NetAspGene/fasta.php

[21] Kong, Q., Liu, Z., Tang, X., Yang, Z., Fu, Y., & Li, H. (2018). The Computation of the Barrier Tree for BHG of RNA Folding Structure. *2018 14th International Conference on Computational Intelligence and Security (CIS)*, 6-9.