RNA 2D Structure Prediction: A review

Reçu le 14/02/2016– Accepté le 20/10/2016

CHEHILI[*] Hamza, HAMIDECHI M. Abdelhafid.

[*] Université frères Mentouri Constantine 1. Faculté des sciences de la Nature et de la Vie. Département de Microbiologie.

Résumé

L'ARN, une macromolécule assurant plusieurs fonctions biologiques : Traduction des gènes en protéines, régulation de l'expression génique, structure 3D et fonction des ARN, etc. Dans ce travail, nous allons passer en revue la prédiction des structures secondaires des ARN par programmation dynamique en s'appuyant sur l'algorithme classique de Nussinov. Nous avons pris en compte quatre possibilité de liaisons entre les nucléotides formant la chaîne polymérique de l'ARN (liaisons canoniques GC, CG, AU UA, liaisons wobble : GU ou UG, etc.). Le programme testé dans ce travail montre que l'algorithme dévelppé prédit correctement les différentes paires de bases qui entrent dans la structure 2D de l'ARN.

Mots clés *: ARN, structure secondaire, algorithme de Nussinov, programmation dynamique.*

Abstract

RNA, a macromolecule that provides several biological functions: gene translation into proteins, regulation of gene expression, prediction of 3D structure and RNA function, etc. In this work, we will review the prediction of RNA secondary structures by dynamic programming based on the classical Nussinov algorithm. We took into account four possible links between the nucleotides forming the RNA polymer chain (canonical GC, CG, AU AU bonds, wobble bonds: GU or UG, etc.). The program tested in this work shows that the developed algorithm correctly predicts the different base pairs that enter the 2D structure of the RNA.

Keywords *: RNA, Secondary structure, Nussinov algorithm, Dynamic programming.*

ملخص

الـRNA جزيئ يوفر العديد من الوظائف الحيوية كترجمة الجينات إلى بروتينات و تنظيم التعبير الجيني و البناء ثلاثي الأبعاد إلخ. في هذا العمل نراجع التنبؤ للبنية ثنائية الأبعاد لـRNA من خلال البرمجة الديناميكية المستندة إلى خوارزم Nussinov . أخذنا بعين الإعتبار الأنواع الممكنة من الروابط بين النكليوتيدات المكونة للحمض الريبي (روابط Watson & Crick ، و روابط GU و UG . و يظهر البرنامج الذي تمّ إختياره في هذا العمل أن الخوارزم المستعمل تنبأ بشكل صحيح بالأزواج القاعدية المختلفة التي تدخل في البنية ثنائية الأبعاد للحمض النووي الريبي.

الكلمات المفتاحية : RNA، *البنية ثنائية الأبعاد ، خوارزم* Nussinov ، *البرمجة الديناميكية.*

The A (adenine), C (cytosine), G (guanine) and U (uracil) nucleotides specify the linear structure of the RNA strand. RNA molecule can construct a double helical structure like DNA molecule by folding of complementary nucleotides. The interactions are observed between A-U and G-C pairs that are called Watson-Crick pairs and G-U, called wobble base pair. The one initial strand becomes a double helical strand because of hydrogene interactions between the different complementary pairs of these nucleotides.

Considering the fourth nucletides in RNA molecule, there is 16 possibilities to construct a pair (AU, AC, AG, AA, UA, UC, UG, UU, CA, CU, CG, CC, GA, GU, GC, GG), however, six possibilities are observed in the native RNA molecules: AU, GC, GU, UA, CG and UG. The other pairs (non canonical) are considered as mismatches. These base pairs are very stable except GU and UG pairs. Because of their low stability caused by the low thermodynamic energies, these two pairs are not stable (Savill *et al.*, 2001).

RNA molecules have a large set of biological functions (mRNA, tRNA, miRNA, …) and the prediction of their 2D structures can help in the understanding of different mechanisms of their interactions in cell environment (*in vivo* functions). The RNA molecules fold to form secondary structures owing to nucleotides complementarities.

The RNA 2D prediction *“is the process by which a linear ribonucleic acid (RNA) molecule acquires secondary structure through intra-molecular interactions. The folded domains of RNA molecules are often the sites of specific interactions with proteins in forming RNA–protein (ribonucleoprotein) complexes.”* (Leppek K. *et al.*, 2017) The 2D RNA structure is predicted from primary structures, using different prediction algorithms.

Overall, there are two major methods for predicting the secondary structure of RNAs : i: Comparative sequence analysis and ii : thermodynamic optimization or minimum free energy method (Zuker M. 1989). In the two cases, the basic motifs of 2D RNA molecules are determined (Fig. 1). Finally, RNA 2D can be considered as a set (conglomeration) of these smaller structures.

The motifs are the result of three fundamental structures:

- stems : regions of RNA with complementary base pairs ; Stacking regions
- loops : unpaired structures, at least four nucleotides long (Hairpin loops, Multiloops (bufurction), Bulges, internal loop,
- free nucleotides.

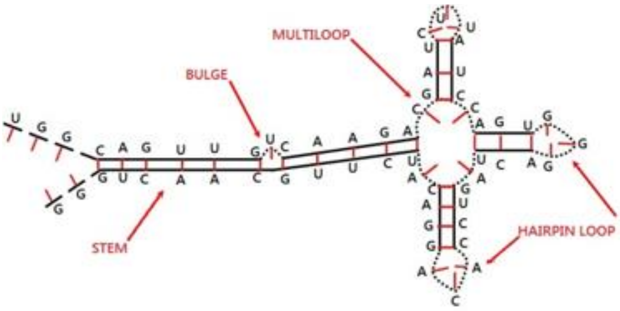


Figure 1 : Example of basic motifs in RNA 2D structures.

FUNDAMENTALS

The four base types construct the basic string of the RNA with a distinguishable 5’-P and 3’-OH ends. Primary structure is followed covalent bonds between nucleotides. Because of the presence of 2’-OH group in ribose sugar, the nucleotides of RNE molecules build up its biological anunctional structure (Sponer J. E. *et al.* 2005)

The secondary structure of the RNAs is the direct result of the interactions between the complementary pairs, keeping as much as possible the free energy of the RNA molecule. The four nucleotides try to achieve a maximum stability for the molecule by reducing its free energy. All of the motifs (Stems and Loops) produce energy that contributes to the overall molecule stability.

We consider that the RNA molecule is a set of successive nucleotides (residues): S={r1, r2, r3, … , rN). The sequence S is a string of characters: S={a,c,g,u}. All complementary pairs are (x,y) = (AU), (UA), (CG), (GC) and (GU) or (UG) which are not treated as a real basepair (Akutsu T., 2000). Moreover, pseudo-knots are usually excluded.

The 2D RNA molecule is defined by a set of energy values of each complementary basepair. And so, complementary regions (stems) are predicted with low energies. The base pairs are shown in tables 1 and 2.

Table 1: Free energy values for stacking base pairs (Burkowski F. J., 2009)

	A/U	C/G	G/C	U/A	G/U	U/G
A/U	-0.9	-2.2	-2.1	-1.1	-0.6	-1.4
C/G	-2.1	-3.3	-2.4	-2.1	-1.4	-2.1
G/C	-2.4	-3.4	-3.3	-2.2	-1.5	-2.5
U/A	-1.3	-2.4	-2.1	-0.9	-1.0	-1.3
G/U	-1.3	-2.5	-2.1	-1.4	-0.5	+1.3
U/G	-1.0	-1.5	-1.4	-0.6	+0.3	-0.5

Table 2 : Free energy values for Loops

Number of bases	1	5	10	20	30
Hairpin	…	4.4	5.3	6.1	6.5
Internal	…	5.3	6.6	7.0	7.4
Bulge	3.9	4.8	5.5	6.3	6.7

DYNAMIC PROGRAMMING

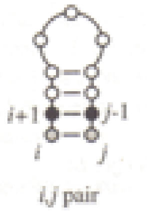
Dynamic programming (DP) solves a large problems encountered in secondary structure determination by using some combination. The principal objective of dynamic programming is to characterize an RNA 2D structure with an optimal solution (a minimum free energy). With DP we can predict 2D structures of RNA molecules with and without pseudoknots. The basic algorithm for DP is without pseudoknots and it was first described by Nussinov (1979) and modified by Zuker *et al.* in 1981 by using the thermodynamic calculation. However, these two algorithms are $O(n^3)$ time and $O(n^2)$ space, where n is the length of the RNA sequence (Zhao C. and Sahni S., 2017).

The classic Nussinov's algorithm is:

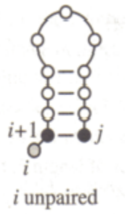
S = x1 x2 x3 … xn ; S is the RNA sequence = {a, c, g, u}, xi are the nucleotides and n is the sequence length. The set of complementary pairs is called M = {(i,j)/1≤i<j≤n}.

The DP Nussinov method is based on these four possibilities:

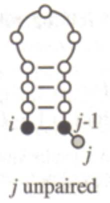
i - Insert a pair of complementary residues i, j for subsequence i + 1, j - 1:



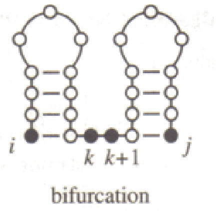
ii - Match the residue at position i with the best position for sub-sequence i+1, j



iii -Match the residue at position j with the best position for sub-sequence i, j-1



iv - Combine two optimal substructures i,k and k+1,j



We consider δ (i,j) = 1 if the two residues i and j are complementary base pair (AU, CG or GU) ; else δ (i,j) = 0. We calculate, recursivly, the scores γ of (i,j) pairs which can be formed into the subseuence xi … xn :

$$\gamma(i,j)=\max\begin{cases}\gamma(i+1,j),\\ \gamma(i,j-1),\\ \gamma(i+1,j-1)+\delta(i,j),\\ \max_{i<k<j}[\gamma(i,k)+\gamma(k+1,j)].\end{cases}$$

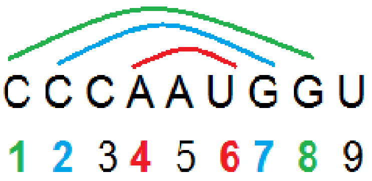
For example, consider the following sequence: CCCAAUGGU

	j	1	2	3	4	5	6	7	8	9
i		C	C	C	A	A	U	G	G	U
1	C	0	0	0	0	0	1	2	3	3
2	C	0	0	0	0	0	1	2	3	3
3	C		0	0	0	0	1	2	2	2
4	A			0	0	0	1	1	1	2
5	A				0	0	1	1	1	1
6	U					0	0	0	0	0
7	G						0	0	0	0
8	G							0	0	0
9	U								0	0

The value of 3 in the case i=1 and j=9 indicates that the ris three matchs, i.e. there is three base pairs which are complementary maximum. The case i=5 and j=6 (A-U) is the first case for which the score is equal to 1 for the first time considering the direction of course of the matrix is made diagonally:

γ(i+1,j-1) + δ(i,j)
=γ(6,5) + δ(5,6)
=0 + 1 = 1

The secondary structure of this sequence is :



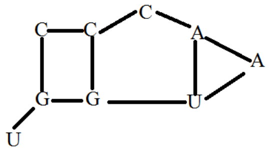
This folding can be represented by :

((. (.))) .

or :

2D = {(1,8) , (2,7) , (4,6)}

or :



Nussinov’s algorithm Implementation

In this section we present Nussinov’s algorithm implementation as a web application using the platform 'Django Python' (Dauzon, S. *et al.* 2016). The application is available in this URL :

https://shielded-chamber-88334.herokuapp.com/.

Based on MVC design pattern (Pop, D. P. and Altar, A, 2014), the application contains the following files:

- The views.py file contains the algorithm implementation
- The forms.py file contains the form used to
- The urls.py file specifies which view is called for a given URL.
- The base.html file is an HTML template that describes the design of the application interfaces.
- The home.html file contains the HTML code of the home page of the application.
- The Nussinovresult.html file is the page of the algorithm results.

The listing 1 presents a part of the views.py code.

```
    while (h>0):
        i=l[h-1][0]
        j=l[h-1][1]
        h=h-1
        p=(i,j)
        s.append(p)
        if (i>=j):
            if (h==0):
                break
            i=l[h-1][0]
....
            else:
                l[h]=p
            h=h+1
        elif y[i,j-1] == y[i,j]:
......
    for i in range(len(seq1)):
        for j in range(len(seq1)):
            matrice[i,j]={"valeur":y[i,j], "i":i, "j":j, "ij": (i,j), "car": seq1[i]}

    structure=list()
    for i in range(len(f)):
         m={"base1": seq2[f[i][0]], "base2": seq2[f[i][1]] }
         structure.append(m)
    if request.POST:
        return render(request, 'lucinov/lucinovresult.html', {'matrice': matrice, 'structure':
structure, 'f': f, 's': s, 'seq2': seq2, 'seq1': seq1, 'numberLines':numberLines,
'numberColumns':numberColumns})
    else:
        return render(request, 'lucinov/acceuil.html', {'form': form} )
```

```
.......
from numpy import *
.....
def lucinov(request):
    form = LucinovForm(request.POST or None)
    def w(i,j):
        if ((seq1[i]=='C' and seq1[j]=='G' ) or (seq1[i]=='G' and seq1[j]=='C' )or
(seq1[i]=='U' and seq1[j]=='A' )or (seq1[i]=='A' and seq1[j]=='U')):
            return 1
        else:
            return 0
    ........
        while j<(len(seq1)):
          i=0
          while i<(len(seq1)) and j<(len(seq1)):
             f=zeros(4)
             f[1]=y[i,j-1]
             f[0]=y[i+1,j]
             f[2]=y[i+1,j-1]+w(i,j)
             m=zeros(j-i-1)
             h=0
             if len(range(i+1,j))>0:
                 k=i+1
                 while k in range(i+1,j):
                     m[h]=y[i,k]+y[k+1,j]
                     h=h+1
                     k=k+1
                 f[3]=max(m)
             if(i==0 and j==1):
                  print(i,j, max(f))
             y[i,j]=int(max(f))
    p=(0,len(seq1)-1)
 ....
```

The listing 2 presents the home.html code.

```
{% extends "base.html" %}
{% load bootstrap3 %}
{% block contentform %}
<form class="form-inline" action="{% url "lucinov" %}" method="post">
 {% csrf_token %}
<div class="form-group">
  {% bootstrap_form form %}
 </div>
 <button type="submit" class="btn btn-default">Calculer</button>
</form>
{% endblock %}
```

```
{% extends "base.html" %}
{% load bootstrap3 %}
{% block title %}Resultat de l'algorithme lucinov !{% endblock %}
{% block contentform %}
    <center>
    <div class="col-md-8">
            <h4>La matrice</h4>
    <table class ="table" style = "text-align:center" width="50%" center>
            <tbody>  <tr>
             <td>  </td>
      {% for car in seq1 %}
                        <td> {{car}} </td>
            {% endfor %}     </tr>
    {% for ligne in matrice %} <tr>
       <td> {{ligne.1.car}} </td>
            {% for case in ligne %}
```

The listing 3 presents the lucinovresult.html code.

```
{% if case.i > case.j %}
                        <td>  </td>
                {% else %}
                        {% if case.ij in f or case.ij in s %}
                                <td class="success">{{ case.valeur}} </td>
                        {% else %}
                                <td>{{ case.valeur}} </td>
                        {% endif %}
                {% endif %}
        {% endfor %}
        </tr>
   {% endfor %}
        </tbody>
   </table>

</div>

<div class="col-md-4">
        <h4>Le schema</h4>

    {% for case in structure %}
```

REFERENCES

1. Savill, N., D. Hoyle, and P. Higgs. 2001. Rna sequence evolution with secondary structure constraints: Comparison of substitution rate models using maximum likelyhood methods. *Genetics*, **157**: 399-411.
2. Zuker M. 1989. On finding all suboptimal foldings of an RNA molecule. *Science*. **7** (244) : 48-52.
3. Akutsu T. (2000). Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*. **104** : 45-62
4. Sponer J. E. Spackova N., Leszczynski J., Sponer J. (2005). Principles of RNA base pairing : Structures and energies of the trans Watson-Crick/sugar edge base pairs. *J. Phys. Chem. B*, **109** (22) : 11399–11410.
5. Burkowski F. J. (2009). *Structural Bioinformatics: An Algorithmic Approach*. CRC Press. Taylor &
6. Francis Group.ISBN :13-978-1-4200-1178-1. 429p.
7. Palkovsky M., Bielecki W. (2017). Parallel tiled Nussinov RNA folding loop nest generated using both dependence graph transitive closure and loop skewing. *BMC Bioinformatics*. **18** : 290-299.
8. Nussinov R., Pieczenik G., Griggs JR., Kleitman DJ. (1978). Algorithms for loop matchings. *SIAM J Appl Math.*; **35**(1) : 68–82. doi: 10.1137/0135006.
9. Zuker M., Stiegler P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res*. ;9(1):133–48. doi: 10.1093/nar/9.1.133.
10. Zhao C., Sahni S., (2017). Cache and energy efficient algorithms for Nussinov's RNA Folding. *BMC Bioinformatics*. **18** (Suppl. 15) : 518
11. Dauzon, S., Bendoraitis, A., Ravindran, A. (2016). Django: Web Development with Python. Packt Publishing Ltd.
12. Pop, D. P., & Altar, A. (2014). Designing an MVC model for rapid web application development. Procedia Engineering, 69, 1172-1179.