Structure secondaire de l'ARN - BIF7001

Abdoulaye Baniré Diallo *Ph.D.*
Professeur, Département d'informatique
UQAM

La structure primaire

Les mots sur $\{A, C, G, U\}$

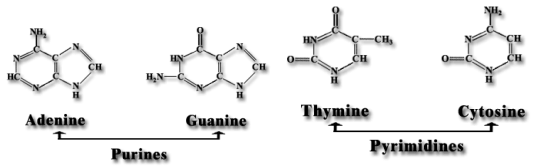
GUCCUCAUAGCUUACAAACCUCAAAGCGCGGCACUG
AAGAUGCCAAGACGGUAACCACCAUACCUGAGGACA
(tRNA-Phe)

Uracile : pyrimidine $\simeq$ Thymine

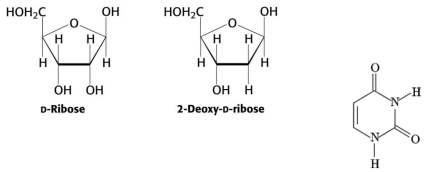
Le sucre des nucléotides = ribose au lieu de désoxyribose dans l'ADN

ARN : simple brin $\Rightarrow$ plus de souplesse dans les structures 2D et 3D

La structure primaire

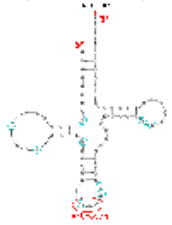


La structure primaire



La structure secondaire

Repliement 2D par création de liens entre paires de bases.

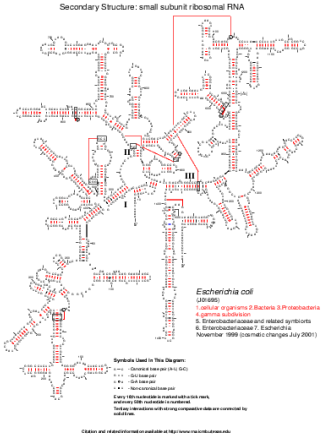


$\left.\begin{matrix} A-U \\ C-G \end{matrix}\right\}$ Watson-Crick

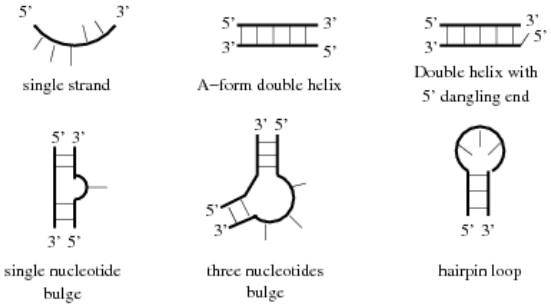
$\left.\begin{matrix} G-U \end{matrix}\right\}$ Wobble

Pas de croisements

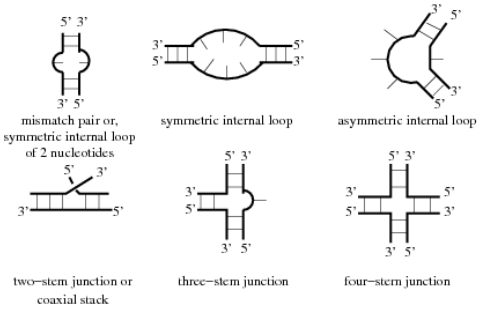
Éléments de structure secondaire



Éléments de structure secondaire



Éléments de structure secondaire



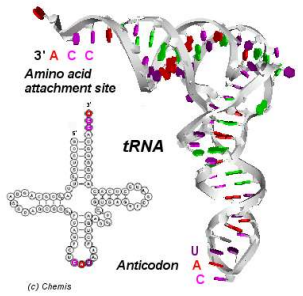
La structure tertiaire

Repliement dans l'espace (3D) de la structure secondaire
Remarque :

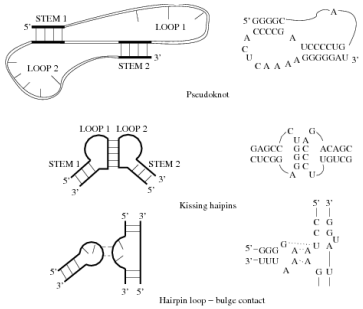
- Les ARN non traduits (ARNt, ARNr, . . .) ont une structure ≪ fixe ≫pour chaque famille
- Les ARNm ont une structure très variable
- La structure est très fortement liée à la fonction

En général, les méthodes ne tiennent pas compte de la structure tertiaire au départ

Structure secondaire vs. Structure tertiaire



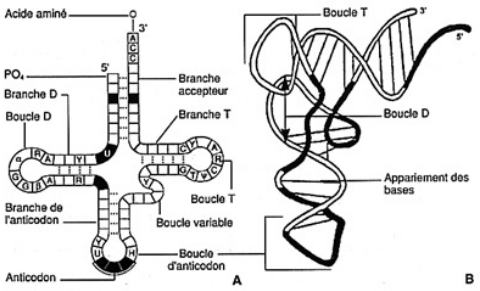
Éléments de structure tertiaire



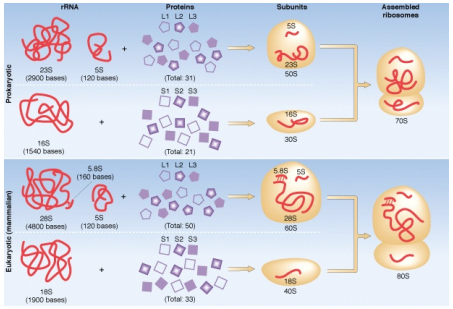
Les types d'ARN

- ARN de transfert
- ARN ribosomial
- ARN messager
- snoARN
- microARN

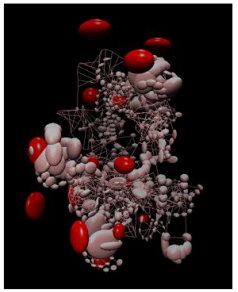
ARNt



ARNr

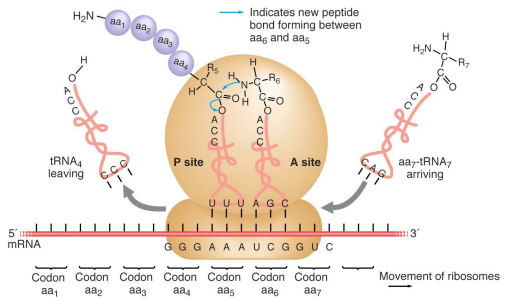


ARNr

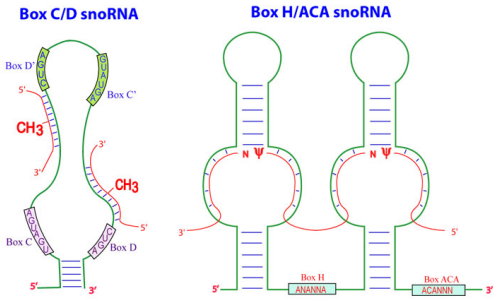


16S rRNA

ARNm

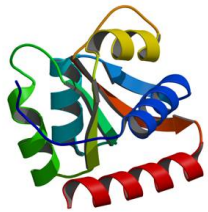


snoARN

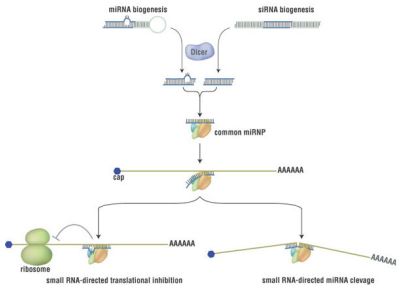


snoARN

Crystal structure of the L7 snoRNA-binding protein from *Methanococcus jannaschii*



micro ARN



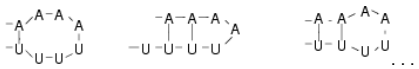
Structure d'ARN et bioinformatique

- Manipulation de structures
  bases de données, sites web, format de fichiers, visualisation
- Prédiction / Repliement
  Structure primaire d'ARN
  ↓ prédiction
  Structure secondaire
- Détection
  À partir d'une séquence d'ADN et d'un schéma de structure secondaire, retrouver dans l'ADN les sous-séquences pouvant se replier selon le schéma donné

Le problème

S = séquence d'ADN
prédire sa structure secondaire

Il y a un nombre exponentiel de repliements possibles
Exemple : AAAAUUUU



Lesquels sont « censés » ?

Un critère de choix : l'énergie

Une paire de bases renforce la stabilité de la structure : il faut beaucoup d'énergie pour la casser
Une boucle déstabilise la structure

Une structure stable demande beaucoup d'énergie pour être modifiée
⇒ on cherche à replier en une structure stable du point de vue énergétique

Plus précisément

Une paire de bases « cachée » dans un gros groupe de paires de base est protégée et dure à casser

Une paire de bases adjacente à une boucle est plus facile à casser

Une paire de bases adjacente à une grosse boucle l'est encore plus

Technique :la programmation dynamique

Premier critère : on essaie de maximiser le nombre de paires de bases en tenant compte du fait que les liaisons A-U et G-C sont très stables, les liaisons G-U sont stables et les autres ne sont pas stables.

Avant de passer au calcul : il faut travailler encore sur la partie ≪ modèle ≫, à savoir déterminer un critère de stabilité.

Algorithme de Nussinov

Principe : calculer le repliement qui maximise le nombre de paires de bases (approximation du maximum d'énergie)

Programmation dynamique :
1. Calcul d'un tableau W : $W_{i,j}$ = nombre maximal de paires de bases parmi tous les repliements possibles du segment $S[i..j]$
$\Rightarrow W_{1,n}$ = nombre de paires de bases d'une structure optimale
2. Calcul d'un chemin dans W pour en déduire une structure optimale.

Algorithme de Nussinov

Calcul de W : programmation dynamique

1. Cas de base : L = taille minimale d'une boucle
$W_{i,j} = 0$ si $j \leq i + L$

2. Récursion : 4 cas pour le calcul de $W_{i,j}$. On suppose $W_{k,l}$ connu

pour $\begin{cases} k = i, & l < j \\ k > i, & l = j \\ k > i & l < j \end{cases}$

Algorithme de Nussinov

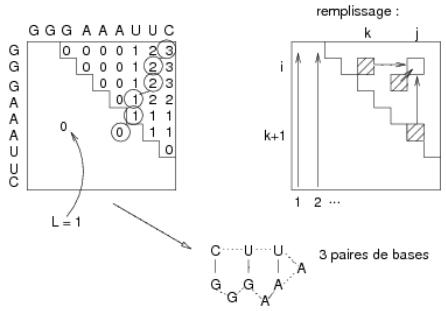
a). i et j forment une paire de bases : $W_{i,j} = 1 + W_{i+1,j-1}$

b). i et j ne sont dans aucune paire de bases : $W_{i,j} = W_{i+1,j-1}$

c). i (resp. j) est dans une paire de bases mais pas j (resp. i) :
$W_{i,j} = W_{i,j-1}$ (resp. $W_{i,j} = W_{i-1,j}$)

d). i et j sont dans deux paires de bases :
$W_{i,j} = \max_{k \in [i+1, j-2]} \{W_{i,k} + W_{k+1,j}\}$

Algorithme de Nussinov



Algorithme de Nussinov

```
Algorithme de Nussinov et al. (repliement d'ARN)
1. Calcul de la matrice W
Pour j de 1 à n faire
    Pour i de 1 à n-j+1 faire
      Si (j ≤s i+L) alors // L longueur minimale d'une boucle
        W[i,j] := 0;
      Sinon
        w := W[i,i+1] + W[i+2,j];
        Pour k de i+2 à j faire
          Si W[i,k] + W[k+1,j] > w alors
            w := W[i,k] + W[k+1,j];
        W[i,j] := MAX{
                    W[i+1,j],
                    W[i,j-1],
                    δ(i,j) + W[i+1,j-1],
                    w

                }
```

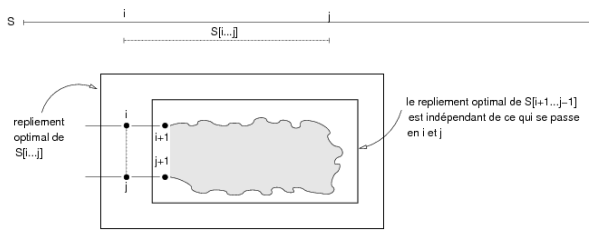
$\delta(i,j) = 1$ si $W[i]$ et $W[j]$ peuvent former une paire de bases, 0 sinon

Algorithme de Nussinov

```
2. Calcul des paires de bases d'une structure secondaire
Soir P une pile vide;
Empiler (1,n) dans P;
Tant que P n'est pas vide faire
    Soit (i,j) le sommet de P;
    Dépiler (i,j) de P;
    Si i >= j ne rien faire;
    Sinon Si W[i+1,j-1] + delta(i,j) = W[i,j]
       Enregistrer (i,j) comme paire de base de la structure secondaire;
       Empiler (i+1,j-1) dans P;
    Sinon Si W[i,j] = W[i+1,j]
       Empiler (i+1,j) dans P;
    Sinon Si W[i,j] = W[i,j-1]
       Empiler (i,j-1) dans P;
    Sinon
       Pour k de i+1 à j-1 faire
          Si W[i,k] + W[k+1,j] = W[i,j] alors
             Empiler (k+1,j) dans P;
             Empiler (i,k) dans P;

             Sortir de la boucle Pour;
```

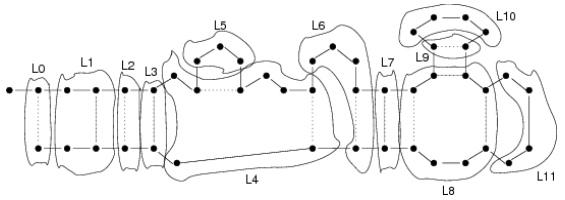
Principe de la programmation dynamique



L'algorithme MFOLD

- Ne prendre en compte que le nombre de paires de bases est insuffisant : une paire de bases isolée stabilise moins que si elle est groupée avec d'autres.
- Mais le principe de la programmation dynamique est intéressant
  ⇒ il faut améliorer le modèle de stabilité énergétique : MFOLD (Zucker)

L'algorithme MFOLD



Idée : décomposer une structure en éléments moins grossiers que de simples paires de bases et associer une énergie à chacun.

$$G(S) = \sum_{i=0}^{11} G(L_i)$$

Éléments structuraux : Loops

- Soit (i,j) une paire de bases
  - la base i' (resp. paire de base (i',j')) est accessible depuis (i,j) si $\forall (k,l)$ paire de bases, on n'a pas $i < k < i'$ (resp. $< j'$) $< l < j$
  - une loop est fermée par (i,j) si toutes ses bases et paires de bases sont accessibles
  - une $k-$loop = une loop à $k-1$ paire de bases
- Hairpin : $1-$loop (L_5, L_{10}, L_{11})
- BasePairs : $2-$loop fermée par (i,j) et avec une seule paire de bases (i',j') : $i' = i+1$ et $j' = j-1$ (L_2, L_3, L_7, L_9)

Éléments structuraux : Loops

- Bulge : $2-$loop $\{(i,j),(i',j')\}$ telle que : $(i' = i+1, j' < j-1)$ ou $(i' > i+1, j' = j-1)$ (L_6)
- InteriorLoop : $2-$loop $\{(i,j),(i',j')\}$ telle que : $i' > i+1, j' < j-1)$ (L_1)
- MultiLoop : $k-$loop, pour $k \geq 3$ (L_4, L_8)
- Stem (ou Stack) : suite de BasePairs $(L_2 L_3)$

Éléments structuraux : Loops

Energie d'une hairpin (exemple de calcul)

Exemple :

1. Loop penalty (loop size=4) +5.60 kcal/mol
2. Stacking GC/AU -2.20 kcal/mol
3. Tetraloop bonus -3.00 kcal/mol

Total : +0.40 kcal/mol

Éléments structuraux : Loops

Energie associée à une loop : le cas des hairpins

Soit (i, j) la paire de bases fermant une hairpin

$$
\begin{aligned}
e_h = \; & e_h^1(i,j) && \text{influence de la longueur de la loop} \\
+ \; & e_h^2(i+1,j-1) && \text{influence du premier mismatch} \\
+ \; & e_h^3(j-i) && \text{terme correctif si la partie terminale} \\
& && \text{a 3 ou 4 bases}
\end{aligned}
$$

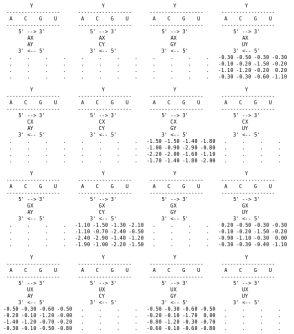
Fichiers

Fichier `loop` de MFOLD
Énergie de déstabilisation d'une loop à 37 degrés C (kCal/mol)

SIZE	INTERNAL	BULGE	HAIRPIN
1	.	3.80	.
2	.	2.80	.
3	.	3.20	5.70
4	1.70	3.60	5.60
5	1.80	4.00	5.60
6	2.00	4.40	5.40
7	2.20	4.60	5.90
8	2.30	4.70	5.60
9	2.40	4.80	6.40
10	2.50	4.90	6.50
11	2.60	5.00	6.60
12	2.70	5.10	6.70
13	2.80	5.20	6.80
14	2.90	5.30	6.90
15	3.00	5.40	6.90
16	3.00	5.40	7.00
17	3.10	5.50	7.10
18	3.10	5.50	7.10
19	3.20	5.60	7.20
20	3.30	5.70	7.20
21	3.30	5.70	7.30
22	3.40	5.80	7.30
23	3.40	5.80	7.40
24	3.40	5.80	7.40
25	3.50	5.90	7.50
26	3.50	5.90	7.50
27	3.60	6.00	7.50
28	3.60	6.00	7.60
29	3.60	6.00	7.60
30	3.70	6.10	7.70

Fichiers

Fichier `tstackh` de MFOLD
Hairpin loop : Enthalpies selon les mismatchs terminaux et les paires de bases à 37 degrés C (kCal/mol)



Fichiers

Fichier `tloops` de MFOLD
Tetraloops : Énergie à 37 degrés C (kCal/mol)

```
Seq    Energy
------------
GGGGAC -3.00
GGUGAC -3.00
CGAAAG -3.00
GGAGAC -3.00
CGCAAG -3.00
GGAAAC -3.00
CGGAAG -3.00
CUUCGG -3.00
CGUGAG -3.00
CGAAGG -2.50
CUACGG -2.50
GGCAAC -2.50
CGCGAG -2.50
UGAGAG -2.50
CGAGAG -2.00
AGAAAU -2.00
CGUAAG -2.00
CUAACG -2.00
UGAAAG -2.00
GGAAGC -1.50
GGGAAC -1.50
UGAAAA -1.50
AGCAAU -1.50
AGUAAU -1.50
CGGGAG -1.50
AGUGAU -1.50
GGCGAC -1.50
GGGAGC -1.50
GUGAAC -1.50
UGGAAA -1.50
```

Énergie

- nécessaire à la stabilité de la structure à une température donnée
- données stockées dans des fichiers distribués avec le logiciel MFOLD (modifiables)
- autres loops :
  - stack : $e_s(i, j)$
  - bulge et interior loop : $e_{bi}(i, j, i', j')$

Énergie d'empilement (stacking)

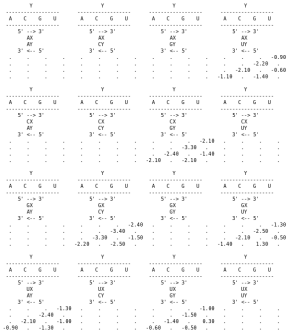
- Les énergie d'empilement sont données en 16 (4x4) tableaux de 16 (4x4) nombres
- Par convention, A, C, G, T/U correspondent à 1, 2, 3 et 4 respectivement

Pour un empilement : 5' -WX- 3' / 3' -ZY- 5' , l'énergie correspondante est dans le tableau de la Wième ligne et la Zième colonne, et dans ce tableau, à la Xième ligne et la Yième colonne. Par exemple, pour W=1 et Z=4 :

```
------------------
 A    C    G    U
------------------
    5' --> 3'
       AX
       UY
    3' <-- 5'
 .     .     .   -0.90
 .     .   -2.20   .
 .   -2.10   .   -0.60
-1.10  .   -1.40   .
```

Énergie d'empilement (stacking)

Fichier `stack` de MFOLD
Enthalpies d'empilement à 37 degrés C (kCal/mol)



Énergie pour les Bulges et Interior loops

```
        C   C
5' – C         G – 3'
3' – G         C – 5'
```

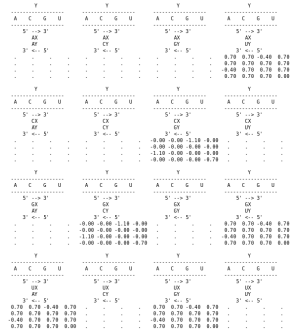
Exemple : U C A

1. Internal loop energy penalty (loop size = 5) : +1.8 kcal/mol
2. Terminal stacking energies for the mismatched base pairs
   adjacent to CG base pair (CG/CU) : 0.0 kcal/mol
   adjacent to GC base pair (CG/AC) : 0.0 kcal/mol
3. For non-symmetric interior loops, there is an asymmetric loop penalty : 0.5 kcal/mol

Total : +2.3 kcal/mol

Énergie pour les Bulges et Interior loops

Fichier `tstacki` de MFOLD
Interior Loops : Enthalpies selon les mismatchs terminaux et les paires de bases à 37 degrés C (kCal/mol)



MFOLD : L'algorithme

- Principe similaire à Nussinov, mais on a 2 tableaux : W et V
  $W_{i,j}$ = énergie (minimum) du repliement optimal de $S[i..j]$
  $V_{i,j}$ = idem mais quand i et j forment une paire de bases ensemble
- Étape de base de la programmation dynamique :
  $W_{i,j} = V_{i,j} = \infty$ si $j \leq i + L$

MFOLD : L'algorithme

- Récursion :
  - $W_{i,j} = \min\{W_{i,j-1}; W_{i+1,j}; \min_{i<k<j}\{W_{i,k} + W_{k+1,j}\}; V_{i,j}\}$
  - $V_{i,j} = \begin{cases} e_h(i,j) & \text{si } (i,j) \text{ forme une hairpin} \\ \text{ou } e_s(i,j) + V_{i+1,j-1} & \text{si } (i,j) \text{ est une 1-loop} \\ \text{ou } e_{bi}(i,j,i',j') & \text{si } (i,j) \text{ est une 2-loop} \end{cases}$

  (multiloops non incluses, compliquées à traiter)
- Complexité : $\mathcal{O}(n^3)$

MFOLD : Le logiciel

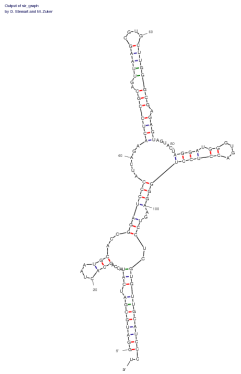
Entrée :

- une séquence
- des paramètres énergétiques
- un ensemble de contraintes
  F 23 87 3 va forcer les paires de bases 23.87, 24.86 et 25.85

Sortie :

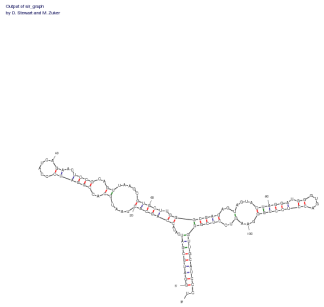
- énergie dot-plot
- RNAML
- un ensemble de structures
- Annotations

MFOLD : Le logiciel



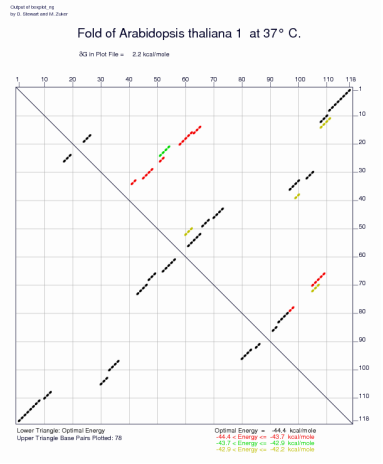
dG = -38.88 [initially -44.40] Arabidopsis thaliana 1

MFOLD : Le logiciel



dG = -41.9 [initially -43.80] Arabidopsis thaliana 1

MFOLD : Le logiciel



Énergie et probabilités : Vienna

Idée : calculer la structure la plus probable (du point de vue énergie) en fonction des paramètres énergétiques choisis.

Mise en œuvre :

- fonction de partition : $\begin{cases} S = & \text{séquence} \\ \mathcal{R} = & \text{espace des repliements de } S \end{cases}$

$$Q_S = \sum_{r \in \mathcal{R}} e^{-G(r)/RT},$$

où $G(r)$ est l'énergie minimum de r, et R, T des constantes connues

Énergie et probabilités : Vienna

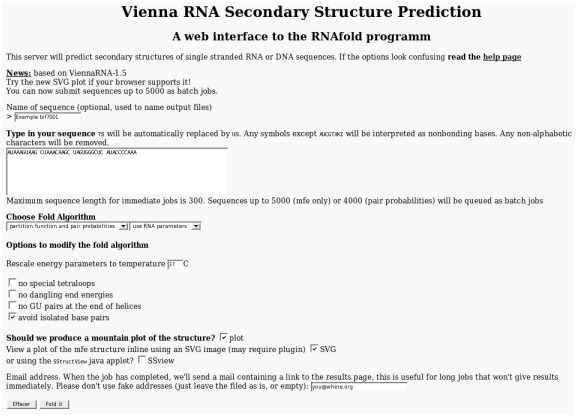
- probabilité d'un repliement r :

$$P(r|S) = e^{-G(r)/RT}/Q_S$$

- calcul de Q_S : « parallèle à MFOLD »
  $G(r) = \sum_{L_i} G(L_i)$
  $e^{-G(r)/RT} = \prod_{L_i} e^{-G(L_i)/RT}$
  $\Rightarrow$ algorithme calqué sur celui de MFOLD

Corollaire : pour une paire de bases (i, j) donnée, on peut calculer la probabilité qu'elle appartienne à un repliement de S.

Énergie et probabilités : Vienna



Vienna RNA Secondary Structure Prediction

A web interface to the RNAfold programm

This server will predict secondary structures of single stranded RNA or DNA sequences. If the options look confusing **read the help page**

News: based on ViennaRNA-1.5
Try the new SVG plot if your browser supports it!
You can now submit sequences up to 5000 as batch jobs.

Name of sequence (optional, used to name output files)
> Exemple bf7001

Type in your sequence Ts will be automatically replaced by us. Any symbols except AUCGTaucgt will be interpreted as nonbonding bases. Any non-alphabetic characters will be removed.

AUAAAGUAAGC CUAAACAAGC UAGUGGGCUC AUACCCCAAA

Maximum sequence length for immediate jobs is 300. Sequences up to 5000 (mfe only) or 4000 (pair probabilities) will be queued as batch jobs

Choose Fold Algorithm
partition function and pair probabilities | use RNA parameters

Options to modify the fold algorithm

Rescale energy parameters to temperature 37 C

- no special tetraloops
- no dangling end energies
- no GU pairs at the end of helices
- avoid isolated base pairs

Should we produce a mountain plot of the structure? plot
View a plot of the mfe structure inline using an SVG image (may require plugin) SVG
or using the structview java applet? SSview

Email address. When the job has completed, we'll send a mail containing a link to the results page, this is useful for long jobs that won't give results immediately. Please don't use fake addresses (just leave the filed as is, or empty): jpu@where.org

Effacer | Fold it

Énergie et probabilités : Vienna

Here are your RNAfold results

RNA parameters are described in
D.H. Mathews, J. Sabina, M. Zucker and H. Turner "Expanded Sequence Dependence of Thermodynamic Parameters Provides Robust Prediction of RNA Secondary Structure", JMB, 288, pp 911-940, 1999

An equivalent RNAfold command line would have been
`RNAFold -p -noLP`

The optimal secondary structure in bracket notation is given below

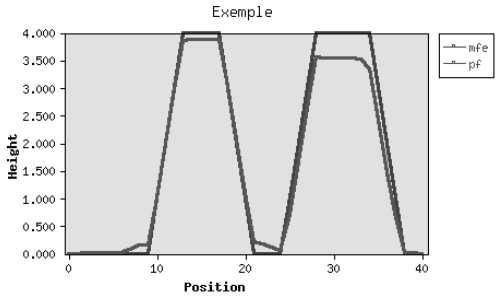
```
> Exemple
AUAAAGUAAGCUAAACAAGCUAGUGGGCUCAUACCCCAAA
.....((((.....))))..((((.....)))).. ( -7.00)
```

The minimum free energy in kcal/mol is given in parenthesis. You may look at the PostScript drawing of the structure in Exemple_ss.ps.

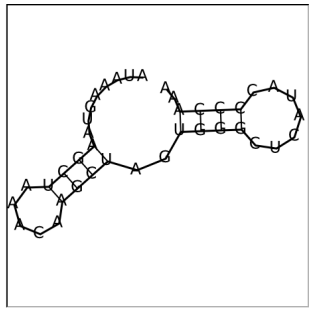
The free energy of the thermodynamic ensemble is **-7.55** kcal/mol
The PostScript *dot plot* containing the base pair probabilities is in Exemple_dp.ps.

The enthalpy of the mfe structure is -90.40 corresponding to a Tm of 63.0C

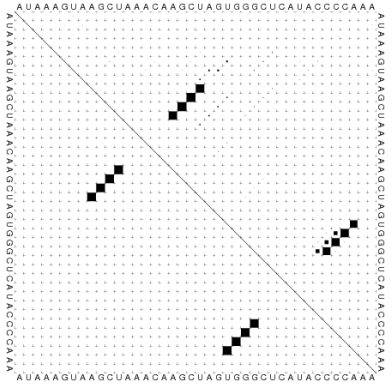
Énergie et probabilités : Vienna



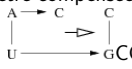
Énergie et probabilités : Vienna



Énergie et probabilités : Vienna



Analyse de covariation de séquences

1. La structure a plus d'importance que la séquence vis-à-vis de la fonction d'un gène : des séquences de même fonction dans plusieurs organismes auront même structure (à peu près) mais des séquences très différentes.
2. La structure secondaire des ARN étant créée par des paires de bases, une mutation d'une base ne modifiant pas la structure devra être compensée par une mutation de l'autre base de la paire : 

COVARIATION.
3. On va donc analyser ces covariations chez plusieurs organismes pour prédire la structure secondaire.

Covariation

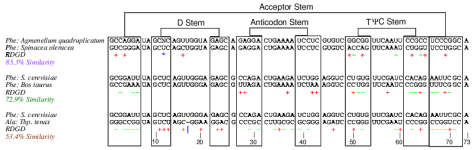


Figure 1. Reddot-greendot examples from tRNA. Symbols used: +: transition; -: transversion; l: deletion; *: ambiguous nucleotide. Experimentally verified helices from the secondary structure are boxed and connected with black lines. Nucleotide position numbers refer to the S.cerevisiae Phe reference sequence. Sequence names are shown as *amino acid:organism*.

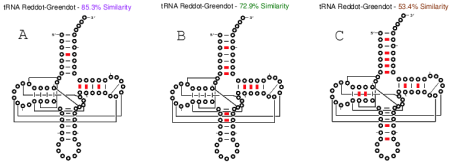


Figure 2. Results of the reddot-greendot analysis shown on tRNA secondary structure diagrams. Base pairs which are predicted with the method are shown with red tick marks. **A.** Sequences with 85.3% similarity. **B.** 72.9% similarity. **C.** 53.4% similarity.

Mutual Information content : MIX

1. Le problème : on a un alignement de séquences dont on suppose qu'elles ont même structure (ou à peu près) : prédire cette structure
2. Calcul : matrice M des scores MIX
   - i, j : colonnes de l'alignement
     $f_i(X)$: fréquence de la base X en colonne i
     (X, Y) : paire de bases AU, UA, GC, CG, GU, UG
     $f_{i,j}(X, Y)$: fréquence de la paire de bases (X, Y) en i et j.
   - $M_{i,j} = \sum_{X,Y} f_{i,j}(X, Y) \log_2 \left(\frac{f_{i,j}(X, Y)}{f_i(X) f_j(Y)} \right)$
   - $M_{i,j} \Rightarrow$ matrice de scores $\in [0, 2]$: plus $M_{i,j}$ est élevé, plus les covariations en colonnes i et j supportent l'hypothèse d'une paire de bases entre ces deux positions.

MIX

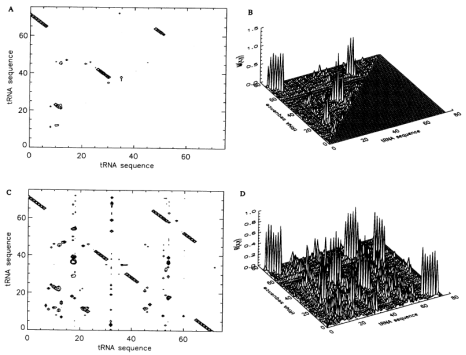


Figure 2. Graphical display of $M(x,y)$ and R values. Only values above 0.2 are displayed on the Contour plots. **A:** Contour plot of $M(x,y)$ values. **B:** Surface plot of $M(x,y)$ values. **C:** Contour plot of R values. The values are determined by taking the values from $M(x,y)$ (shown in part A) and replicating them symmetrically into the other half of the matrix, and then dividing each row by the entropy of the position on the vertical axis. As described in the text, $R_1(x,y) = R_2(y,x)$, so this plot shows both values. If the vertical axis is considered to be position x, then the plot is of $R_1(x,y)$; if the vertical axis is considered to be position y then the plot is of $R_2(x,y)$. Sorting by $R_1(x,y)$ is equivalent to sorting within rows and sorting by $R_2(x,y)$ is equivalent to sorting within columns. **D:** Surface plot of R values. The values are the same as in part C, but displayed as a 3-D plot.

MIX

1

i	j	
A	U	$f_i(A) = 2/3$
C	G	$f_j(A) = 0$
A	G	$f_{ij}(AU) = 1/3$
		$f_{ij}(AG) =$ non défini

$$M_{ij} = 1/3 \log_2 \left(\frac{1/3}{2/3 \cdot 1/3} \right) + 1/3 \log_2 \left(\frac{1/3}{1/3 \cdot 2/3} \right) (\simeq 0.389975)$$

MIX

2

i	j	i	j	i	j	i	j	i	j
A	C	A	U	A	U	A	U	A	U
G	U	A	U	A	U	C	G	C	G
G	A	A	U	G	C	G	U	G	U
A	G	A	U	G	C	G	C	U	A
0		0		1		7/4		2	

aucune covariation pour appuyer l'hypothèse d'une paire de bases

peu de covariation, 1 mutation puis évolution

fortes covariations

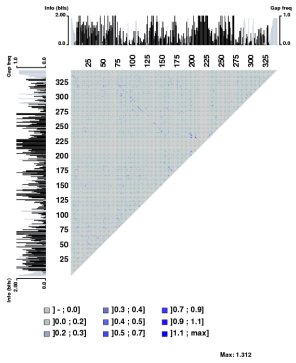
MIX

3 Logiciels
- MatrixPlot (Gorodkin et al.).
- Structure Logo
- cf. CRW

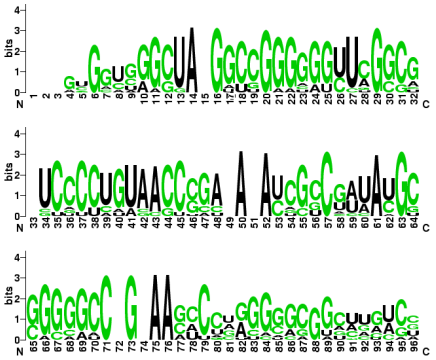
4 Défaut : nécessite un alignement structurel

5 Ne prédit pas mais indique des hélices possibles

MatrixPlot



Structure Logo



Prédiction par analyse des covariations

Algorithme de Parsch et al. (2000)

Calcul en deux étapes : (données = alignement)

1. Calcul d'une matrice BP
   $BP(i,j)$ = type de paire de bases le plus probable entre les colonnes i et j de l'alignement (seuil déterminé par l'utilisateur) : Watson-Cricks, Wooble, aucune, ...
   Identification de séquences de paires de bases probables (hélices) et calcul d'un score LRT pour chacune.
2. Assemblage des hélices en structures, par regroupement des hélices compatibles et calcul d'un score LRT pour chaque structure

Prédiction par analyse des covariations

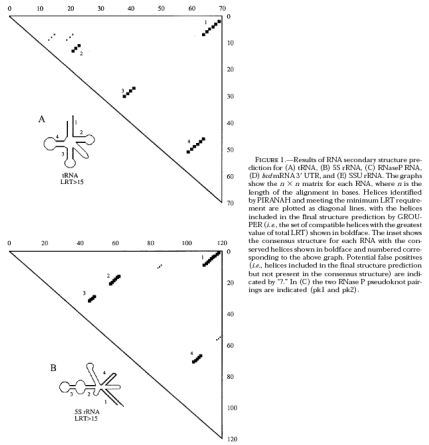


FIGURE 1.—Results of RNA secondary structure prediction for (A) tRNA, (B) 5S rRNA, (C) RNaseP RNA, (D) *bsfmRNA* 3′ UTR, and (E) SSU rRNA. The graphs show the $n \times n$ matrix for each RNA, where n is the length of the alignment in bases. Helices identified by FIRNAH and meeting the minimum LRT requirement are plotted as diagonal lines, with the helices included in the final structure prediction by GREUPER (*i.e.*, the set of compatible helices with the greatest value of total LRT) shown in boldface. The inset shows the consensus structure for each RNA with the conserved helices shown in boldface and numbered corresponding to the above graph. Potential false positives (*i.e.*, helices included in the final structure prediction but not present in the consensus structure) are indicated by "/." In (C) the two RNase P pseudoknot pairings are indicated (pk1 and pk2).

Utiliser covariation **et** énergie minimum

1. Défauts des 2 méthodes :
   - covariation : besoin de nombreuses séquences homologues et divergentes et d'un bon alignement
   - énergie : confiance dans les paramètres thermodynamiques ; pertinence du concept
2. Un exemple d'utilisation conjointe : Construct (Lück et al. 1999)

Utiliser covariation **et** énergie minimum

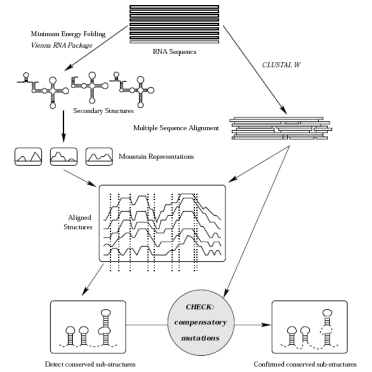
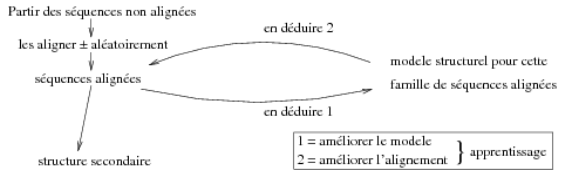


Figure 2: Scheme of the secondary structure analysis of viral genomes.

Modélisation et apprentissage

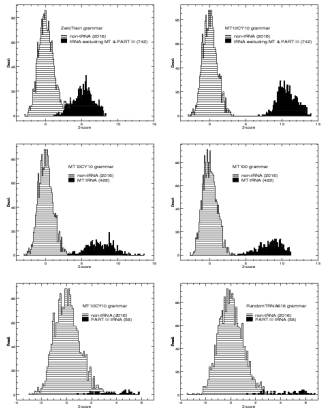
1. Problème avec toute approche basée sur la covariation :
   - pour prédire une structure, il faut un alignement correct des séquences
   - pour aligner correctement, il faut connaître la structure
2. Idée :



Modélisation et apprentissage

3. Technique : modèle = modèle stochastique basé sur une grammaire et un HMM
   Durbin et al. logiciel COVE
   Sakakibara et al. logiciel RNACAD

Modélisation et apprentissage



Modélisation et apprentissage

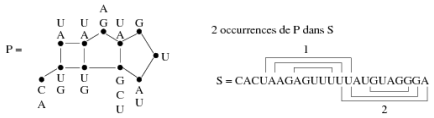
Table 2. Training and multiple alignment results from models trained from the trusted alignments (A models) and models trained from no prior knowledge of tRNA (U models)

Model	Training set	Iterations	Score (bits)	Alignment accuracy
A1415	all sequences (aligned)	3	58.7	95%
A100	SIM100 (aligned)	3	57.3	94%
A65	SIM65 (aligned)	3	46.7	93%
U100	SIM100 (degapped)	23	56.7	90%
U65	SIM65 (degapped)	29	47.2	91%

Détection d'ARN dans une séquence

1 Le problème
- Données : S une séquence d'ADN
  P une description d'une famille de structures secondaires
- Rechercher dans S toutes les sous-séquences pouvant se replier en une structure secondaire décrite par P

Exemple :



Détection d'ARN dans une séquence

2 Deux types de programmes :
- optimisés pour un type d'ARN (ARNt, Introns groupe I, motifs stem-loop) : P fixé
- généraux : l'utilisateur définit P

Deux problèmes bioinformatiques

Définir un motif de structure secondaire assez spécifique, mais pas trop rigide : consensus de ce qui est connu pour cette famille

Rechercher efficacement les occurences de ce motif

On a déjà vu un outil : RNACAD, COVE

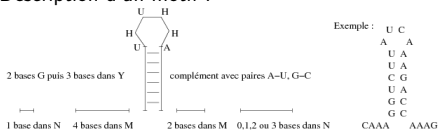
Analogie

Recherche de motifs primaires : KMP,BM, automates, recherche de motifs approchés

mais compliqué par l'emphase sur la structure plus que sur la séquence

Un programme général : PatSearch

1 Description d'un motif :



$r_1 = \{au, ua, cg, gc\}$
`0...1 mmmm` $p_1 =$ `ggyyy u huhh a` $r_1 \simeq p_1$ `mm 0...3`
semblable à une **expression régulière !**

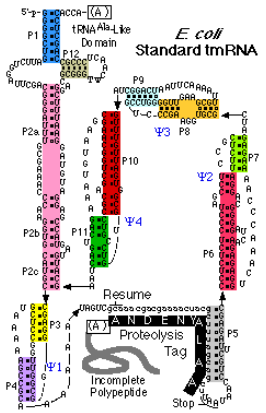
Un programme général : PatSearch

2 types d'éléments :

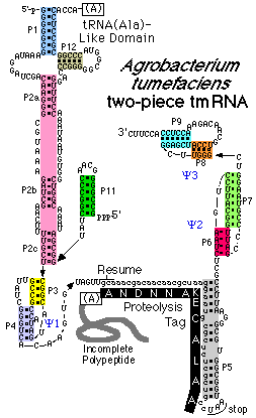
- pairing rules
- pattern units
  - → intervalle `i...j`
  - → séquence `u ; mm ; ggyyy`
  - → complément d'une précédente pattern unit identifiée $r_1 \simeq p_1$

2 La recherche des occurrences : algorithme naïf de recherche de motifs avec backtracking

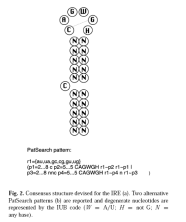
PatSearch



PatSearch



PatSearch

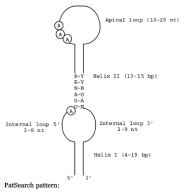


PatSearch pattern:

r1=(au,ua,gc,cg,gu,ug)
r2=(uu,cu,uc,au,ua,gu,gg)
p1=2..3 p2=3 CAGWGH r1-p2 r1-p1 !
p3=2..8 rev_p1=5..3 CAGWGH r1-p4 r1-p3 !

Fig. 2. Consensus structure devised for the IRE (a). Two alternative PatSearch patterns (b) are reported and degenerate nucleotides are represented by the IUB code (W = A/U; H = not G; N = any base).

N = A ou C ou G ou U
W = A ou U
H = A ou C ou G



PatSearch pattern:

r1=(au,ua,gc,cg,gu,ug)
r2=(uu,cu,uc,au,ua,gu,gg)
p1=4...19
p2=2...7 a
uga p3=1...1 p4=rr p5=7...9
p6=0...2 aav p7=5...15
r1–p6[1,1,1] r1–p4 r2–p3 gan
p8=2...9 –p1[1,0,0]

Fig. 3. Consensus structure devised for the Selenocysteine insertion sequence (a). In the corresponding PatSearch pattern (b) two different pairing rules ($r1$ and $r2$) are used for different helices where mismatches and indels are allowed in some cases.

Algorithme de PatSearch

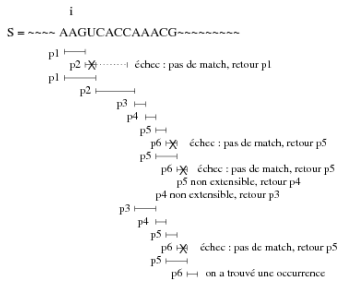
Données : S séquence de m nucléotides ; $p_1, \ldots, p_n$ suite de pattern units

1. Pour i de 1 à m faire
2. $\quad j = i$; $k = 1$; $P = \emptyset$ (pile vide);
3. $\quad l =$ longueur minimale de p_1 ;
4. $\quad$ Répéter
5. $\quad\quad$ Si ($l >$ longueur maximale de p_k) alors
6. $\quad\quad\quad$ Si ($P = \emptyset$) alors $j = m + 1$;
7. $\quad\quad\quad$ Sinon $l =$ Dépiler(P)+1;
8. $\quad\quad$ Sinon Si $S[i \ldots i + l - 1]$ matche p_k alors
9. $\quad\quad\quad$ Empiler (P,l) ; $k = k + 1$; $j = j + l$;
10. $\quad\quad\quad$ Sinon Si ($P = \emptyset$) alors $j = n + 1$;
11. $\quad\quad\quad\quad$ Sinon $l =$ Dépiler(P)+l;
12. $\quad$ jusqu'à ce que $k > n$ ou $j > m$;
13. $\quad$ Si ($k > n$) alors $\ll$ occurrence entre i et $j \gg$
14. $\quad$ Sinon $\ll$ pas d'occurrence en $i \gg$

Complexité : $\prod_{i=1}^{n} |p_i|$

Algorithme de PatSearch

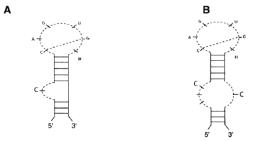
Exemple : $p_1 = 2..3$, $p_2 =$YYAY, $p_3 = 1..2$, $p_4 =$A, $p_5 = 1..2$, $p_6 =$G



Autres programmes

- Palingol : motif = suite d'hélices (pattern units) + contraintes de proximité entre les hélices + contraintes tertiaires
  algo (exponentiel) = liste des hélices puis combinaison
- RNAMotif : motif semblable à PatSearch, algorithme idem.
  Score (GC, erreurs, énergie) : plus spécifique
- COVE (covels) motif = SCFG : grammaire, algo = CYK : programmation dynamique
- El-Mabrouk et Raffinot

RNAMotif



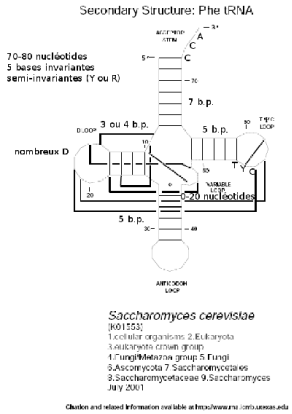

```
parms
wc += gu;
descr
ss( len=3 )
bS( tag='lower_stem', len=3 )
ss( tag='5p_bulge', minlen=1, maxlen=3, tag='5p', seq="cS" )
bS( tag='upper_stem', len=5 )
ss( len=6, seq="'nagug" )
bl(tag='upper_stem')
ss(tag='3p_bulge', minlen=0, maxlen=1)
bl(tag='lower_stem')
ss( len=3 )
score
{
SCORE = 0.0;
if( length(ss(tag="5p")) == 3 && length( ss( tag="3p")) == 1 ){
if( ss(tag="5p") == "tgc" )
        SCORE = 0.5;
else if( ss(tag="5p") == "tac" )
        SCORE = 0.4;
else if( ss(tag="5p") == "tcc" )
        SCORE = 0.3;
else if( ss(tag="5p") == "ttc" )
        SCORE = 0.2;
else if( ss(tag="5p",pos=1,len=1) == "c" )
        SCORE = 0.1;
if( ss(tag="3p") == "c" )
        SCORE += 0.5;
else if( ss(tag="3p") == "t" )
        SCORE += 0.2;
ACCEPT;
}else if( length(ss(tag="5p" )) == 1 && length(ss(tag="3p")) == 0 ){
SCORE = 1.0;
ACCEPT;
}else
REJECT;
```

Recherche d'ARNt : FasttRNA

- Base : la structure des ARNt est suffisamment stable pour un programme très spécifique
- Idée algorithmique :
  1. associer un signal aux bras TΨC et D
  2. rechercher uniquement ces signaux (simples) dans S
  3. pour chaque région de S contenant ces deux signaux, l'examiner en détail (i.e. lentement) pour essayer de la replier en ARNt

Étape 2 : recherche de motifs primaires approchée à l'aide d'un algorithme bit-vecteur Shift-Add.

FasttRNA



FasttRNA

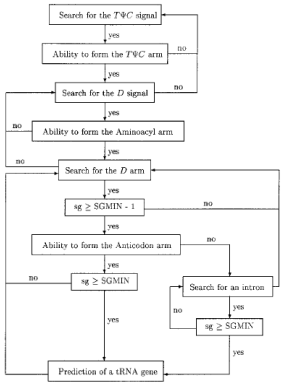


Figure 2. Flow chart of the algorithm.

FasttRNA

Table 4. Parameter and threshold values used in *FAStRNA*

Region[a]	Perfect match[b]	Threshold matc
TΨC signal	At most 1 mismatch	Three mismatches
TΨC arm	base-pairing score >26	Base-pairing score >10 and at least 3 base-pairing
D signal	No mismatch	Two mismatches
Aminoacyl arm	Base-pairing score >36	base-pairing score >18 and at least 4 base-pairing
D arm	Score of the 4 base-pairs >16	Score of the 3 first base-pai or score of the 3 first base- and 4th base-pairing >0
Base 18, 19 and 21	Base 18 = G, base 19 = G and base 21 = A	Other bases
Base 33	T	Other base
Anticodon arm		
Without intron	Base-pairing score >19	Base-pairing score >11
With intron	Base-pairing score >26	Base-pairing score >17

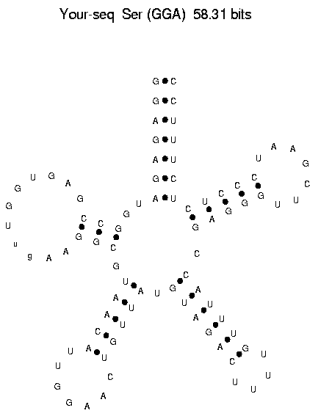
[a] Regions of the tRNA-sequence chronologically analysed by the algorithm.
[b] Each time a condition is verified, the general score *sg* is incremented.
[c] Minimal conditions for accepting a region.

Autres programmes

- tRNAscan : même idée de signal mais approche probabiliste
- tRNAscan_SE : modèle de covariation

Important : modélisation par expression régulière puis adaptation d'algorithmes de recherche de motifs primaires

FasttRNA



Détection d'ARN : résumé

- ARN précis (ARNt, Introns) : programmes dédiés
- Structure connue (ou en partie) : définir un motif, rechercher ses occurrences, examiner les hits et leurs repliements
- Ensemble de séquences disponibles (Rfam ! !) alignées ou non : modèle de covariation
- En général, moins on a d'information, plus la recherche est longue ⇒ intérêt à bien cibler les zones examinées
- Une fois une séquence plausible repérée, la replier aide à la classifier comme ARN ou non
- ARN non-codants : le grand défi