
# Informatique Génomique — ARN

**TD** #1 **02 Avril 2011**

**Rédacteur**: *Stéphane Vialette*

## Objectifs

Les objectifs de ce TP sont :

1. programmer l'algorithme de Nussinov pour le repliement d'ARN, et
2. programmer un algorithme de comparaison de structures secondaires d'ARN (type Zhang - Shasha).

Les exemples donnés ici utilisent le langage python, mais vous êtes libres d'utiliser le langage de votre choix.

## Algorithme de Nussinov

Il s'agit ici de mettre en œuvre l'algorithme de Nussinov (1978) pour calculer le repliement d'un ARN. L'algorithme de Nussinov ne cherche qu'à maximiser le nombre total d'appariements de base dans la structure formée, sans critères énergétiques, ni prise en compte des interactions d'empilements entre paires de bases. (Une version améliorée de cet algorithme de base a été proposée en 1981 par Michael Zuker, qui incorpore les données thermodynamiques et en particulier les interactions d'empilement.)

Vous devez dans votre mise en œuvre séparer le système de score de l'algorithme en lui même. Par exemple, dans l'exemple donné ci-après, la classe `FoldingScoringScheme` se charge du chargement et de la consultation du système de score et la classe `NussinovFolder` est en charge de l'algorithme (indépendamment d'un système de score particulier).

```
if __name__ == '__main__':
    # our reference RNA sequence
    seq = 'CGGAUACUUCUUAGACGA'

    # default scoring scheme
    # G-C=3, A-U=2, G-U=1, others=0
    scoring_scheme = FoldingScoringScheme()
```

```
    # or set some specific score scheme, for example
    # base_pairing = (('G', 'C', 4), ('A', 'U', 3), ('G', 'U', 1))
    # scoring_scheme = FoldingScoringScheme(base_pairing)
    # X-Y = 0 if not explicitly defined
    print 'scoring scheme ...'
    print scoring_scheme

    # folding our RNA sequence
    rna_fold = NussinovFolder(seq, scoring_scheme)
    print 'folding ...'
    print rna_fold
```

Ce programme produit la sortie suivante.

```
$ python ex_fold.py
scoring scheme ...
C-G: 3, A-U: 2, G-U: 1, others: 0
folding ...
score: 14
> C G G A U A C U U C U U A G A C G A
> ( ( ( . . . ) ( . ( . . . ) ) ) ) .
$
```

Il n'est pas demandé de produire toutes les solutions optimales, une est suffisante. Par contre, votre sortie devra se présenter sous forme d'un mot sur l'alphabet $\{(, ), .\}$ superposé à la séquence considérée.

## Comparaison de structures secondaires

Comparer des structures secondaires d'ARN est un problème fondamental en bioinformatique. Les structures secondaires d'ARN sont souvent modélisées par des arbres ou par des séquences arc-annotées. Nous nous intéresserons ici au calcul de la distance d'édition entre deux arbres ordonnés (l'ordre des fils est fixe, de gauche à droite).

Pour faire connaissance avec le sujet, je vous conseille la lecture rapide de la page wikipedia (`http://fr.wikipedia.org/wiki/Distance_d'%C3%A9dition_sur_les_arbres`). Une présentation claire de l'algorithme de Zhang-Shasha est disponible ici (`http://uuu.enseirb-matmeca.fr/~allali/arbre.pdf`), pages 53-55.

Voici un exemple d'utilisation pour le calcul de la distance d'édition (en utilisant nos classes `FoldingScoringScheme` et `NussinovFolder`). La classe `RNATree` est en fait une mise œuvre d'une structure d'arbre ordonné (la méthode `__init__` de la classe `RNATree` prend en argument un repliement (optimal ou non) – une instance de la classe `NussinovFolder` – et construit l'arbre ordonné à partir de celle ci).

```
if __name__ == '__main__':

    # first RNA as tree
    rna_fold1 = NussinovFolder('ACGUACGU', FoldingScoringScheme())
    rna_tree1 = RNATree(rna_fold1)

    # second RNA as tree
    rna_fold2 = NussinovFolder('GCGAACGU', FoldingScoringScheme())
    rna_tree2 = RNATree(rna_fold2)

    # use a default scoring scheme for the edit distance
    scoring_scheme = EditDistanceScoringScheme()

    # compute the edit distance beteen the two RNA trees
    edit_distance = RNAEditDistance(rna_tree1, rna_tree2, scoring_scheme)

    # print the edit operations
    print 'edit distance: ', edit_distance.distance()
    print 'edit operations:'
    for edit_op in edit_distance:
        print edit_op
```