

## Travaux pratiques :

On commencera par **terminer le TP1** si celui-ci n'a pas été fini ...

### RNASubopts

RNASubopts est un programme qui permet la génération de structures sous-optimales comme décrit en cours. On lui fournit une séquence d'ARN  $\omega$  ainsi qu'une tolérance  $\Delta$  sur la sous-optimalité, et il renvoie la liste de toutes les structures secondaires pour  $\omega$  ayant une énergie dans l'intervalle  $[\mathbf{mfe}, \mathbf{mfe} + \Delta]$ , où  $\mathbf{mfe}$  est l'énergie la plus faible d'une structure dans l'ensemble de Boltzmann.

Codez une fonction `runRNASubopts( $\omega, \Delta$ )` qui invoque RNASubopts avec les arguments correspondants, et renvoie sous la forme d'une liste structure/énergie les structures obtenues.

### RNAForester

Faite de même pour le programme RNAForester, aussi décrit en cours et qui compare deux couples séquence/structure d'ARN. La fonction `runRNAForester( $\omega_1, S_1, \omega_2, S_2$ )` invoquera RNAForester avec les paramètres par défaut, et retournera uniquement le score d'alignement.

## Entropie

De nombreuses mesures ont été proposées pour évaluer la stabilité du repliement d'un ARN. On en propose ici une nouvelle, basée sur la distance moyenne dans l'ensemble de Boltzmann. Afin d'approcher cette mesure pour un ARN  $\omega$  donné, on procède comme suit :

- Calculer avec RNAFold -p la valeur  $\mathcal{Z}$  de la fonction de partition.
- Engendrer avec RNASubopts l'ensemble  $\mathcal{S}$  des structures à distance énergétique  $\Delta = 2$  KCal.mol<sup>-1</sup> de la structure optimale.
- Pour chacune de ces structures, calculer le facteur de Boltzmann  $B_S = e^{-E/RT}$ , puis la probabilité de Boltzmann associée  $P_S = B_S/\mathcal{Z}$ .
- Répéter  $M$  fois, en partant d'un ensemble d'échantillons  $V = \emptyset$  :
  - Tirer au hasard (avec la fonction python `random.sample`) un couple de structures  $(S_1, S_2)$ , de probabilité jointe  $W_{S_1, S_2} = P_{S_1} \cdot P_{S_2}$ .
  - Calculer avec RNAForester la distance d'alignement  $D_{S_1, S_2}$ .
  - Ajouter le couple  $(W_{S_1, S_2}, D_{S_1, S_2})$  à l'ensemble des échantillons  $V$ .
- Calculer et renvoyer la distance moyenne pondérée

$$D^* = \frac{\sum_{(w,d) \in V} w \cdot d}{W} \quad \text{où} \quad W = \sum_{(w,d) \in V} w.$$

## Clustering

Proposez une méthode (heuristique) pour classer un ensemble de structures en clusters, en se basant sur le critère de distance deux à deux décrit dans le TP 1. On pourra par exemple partir d'une structure tirée aléatoirement, et ajouter au cluster toutes les structures à distance inférieure à un seuil dont vous choisirez la valeur ... Calculer pour chaque cluster une *structure représentative*, présentant une paire de base  $(i, j)$  si et seulement si celle-ci est présente dans au moins 50% des structures du cluster.

**Benchmark ... suite**

Comparez les performances de l'algorithme de prédiction basé sur la structure représentative du plus lourd cluster (poids = somme des facteurs de Boltzmann) aux prédictions de `RNAfold`. Calculez à la fois la sensibilité et la positive predicted value (PPV). Qu'observez vous ?

**Algorithme d'Akutsu**

Reprenez l'algorithme d'Akutsu pour les pseudo-noeuds simples vus en cours. Vérifier que chaque pseudo-noeuds simple est décrit une et une seule fois par la décomposition. Comparer les structures prédites par des algorithmes de repliement sans (cf TP 1) et avec pseudonoeuds simples.