Vendredi 26 Mars 2004

Démo 9 – Recherche de structures secondaires de l'ARN

Jean-Eudes Duchesne

1. Mini historique des méthodes de recherche

i) Algorithmes pour la recherche exacte.
Le but est de trouver un mot P dans une séquence S sans erreurs.
- Algo naïf, O(n*m)
- Knuth-Morris-Pratt, O(n)
- Boyer-Moore, O(n)

ii) Algorithmes de recherche de motifs simples.
Recherche de motifs tels que des séquences répétées, des séquences miroirs et ainsi de suite. Utilisation d'un arbre de suffixe. Construction de l'arbre se fait en temps O(n). La recherche dans l'arbre peut prendre un temps O(n) ou O(1) tout dépendamment du problème et de la complexité de l'arbre créer.

iii) Algorithmes de recherche approchée (distance d'édition).
- Algo naif, $O(n*m^2)$.
- Smith-Waterman (alignement semi-local), O(n*m).
- Algorithmes numériques (Shift-Add, Shift-Or, Shift-And, Myers). Capable d'atteindre O(n) si p< taille d'un mot machine.

iv) Algorithmes de recherche approchée de structures secondaires.
- Recherche de molécules biologiques spécifiques, O(n).
- Recherche générale, $O(sn^2)$ avec grammaires.
- Modèle d'apprentissage. O(???), mais lent !

2. Classe d'algorithmes de recherche de structures secondaires

Sûr mesure	Générale	Apprentissage
ARNt : tRNAscan, FAStRNA, tRNAscan-se, Bruce, ARAGORN. Intron groupe I : CITRON SnoRNA : snoSCAN	RNAMOT, RNAMOTIF, palingol, patscan, Biosmatch.	Covel, ERpin

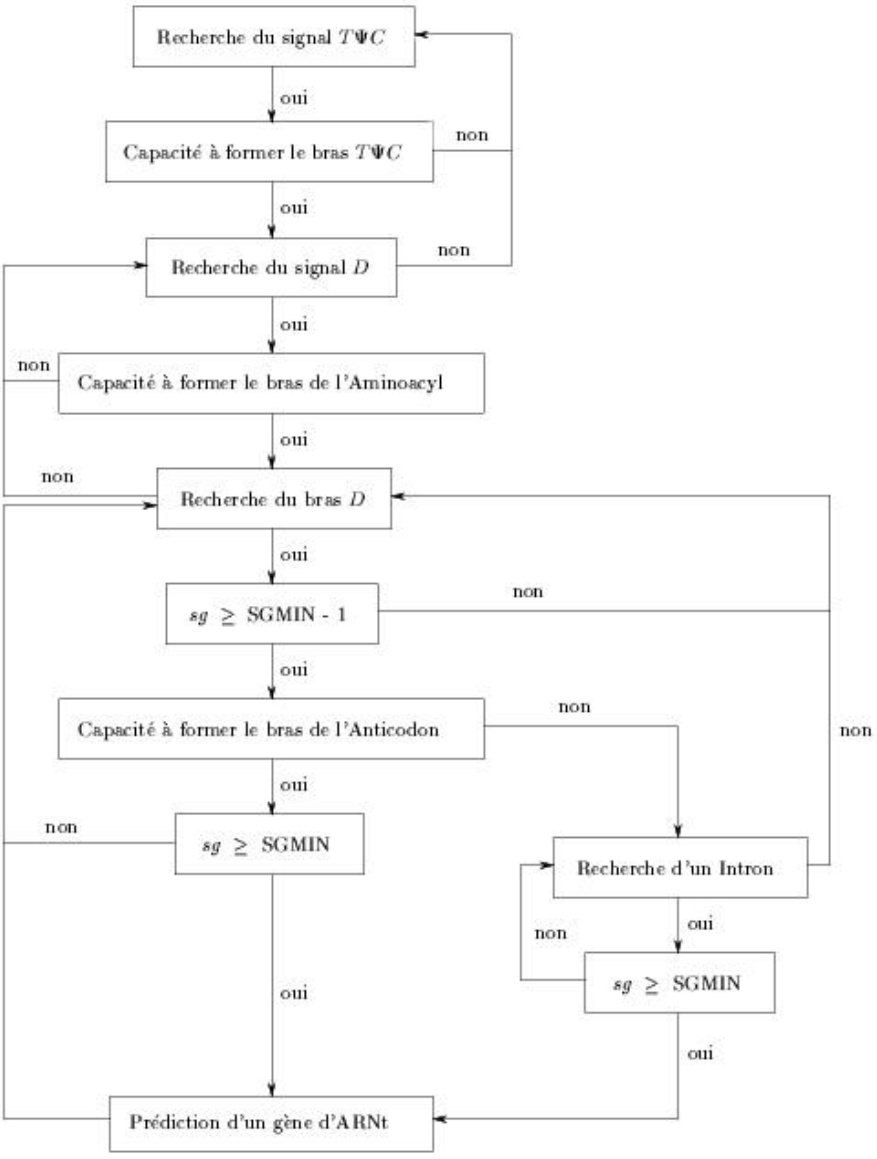
L'approche sûr mesure se veut la plus efficace et la plus rapide des approches. Par contre, celle-ci n'est pas généralisable et très peu flexible. L'idée générale est de rechercher les éléments les plus conservés d'une molécule particulière et d'utiliser ces éléments comme encre pour reconstruire la molécule totale. De façon globale, c'est une approche par satisfaction de contraintes.

L'approche générale utilise plutôt un descripteur pour effectuer sa recherche. Ceci est beaucoup plus flexible puisque c'est l'utilisateur qui entre la séquence à rechercher. Par

contre les résultats dépendent absolument du descripteur. Un bon descripteur donne de bons résultats et inversement pour un mauvais descripteur. Aussi, ces méthodes sont souvent plus lentes puisque des descripteurs avec un grand pouvoir expressif représente un problème de recherche plus complexe que la recherche de motifs conservés. Par contre, ces méthodes permettent à l'usager d'utiliser l'information biologique la plus récente puisque c'est l'usager qui est responsable de mettre à jour le descripteur. Dans le cas de l'approche sûr mesure, l'usager dépend des mises à jours de celui qui maintient le code.

L'approche par apprentissage ressemble à l'approche générale dans le sens où l'algorithme de recherche utilise aussi un descripteur, mais ici l'usager n'est pas responsable de la confection du descripteur. Le programme prends en entrée un ensemble de séquences de la même familles (ex : ARNt de plusieurs espèces différentes) et détermine lui-même les éléments structuraux à rechercher. Dans ce cas, la phase de construction du descripteur est souvent plus longue que la phase de recherche.

3. Exemple sûr mesure : FAStRNA

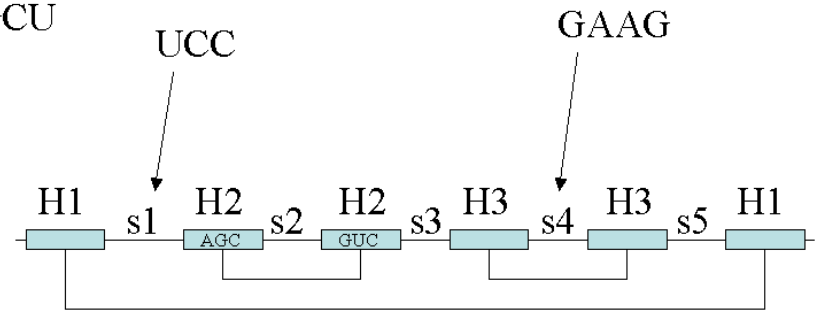


4. Exemple général : RNAMOT et RNAMOTIF

Descripteur RNAMOT :

H1 s1 H2 s2 H2 s3 H3 s4 H3 s5 H1

H1 3:5 0
H2 4:5 1 AGC:GCU
H3 4:5 1
S1 3:6 ucc
S2 5:7
S3 0:3
S4 5:8 gaag
S5 3:5



RNATMOTIF est une version améliorée de RNAMOT considéré comme le programme de référence en ce moment dans la recherche de motifs biologiques de façon générale.

RNAMOTIF :

1. Compilation du descripteur (vérification et construction de l'arbre de recherche).
2. Algorithme de recherche (parcours de l'arbre en profondeur d'abord, backtraking).
3. Attribution du score (optionel).

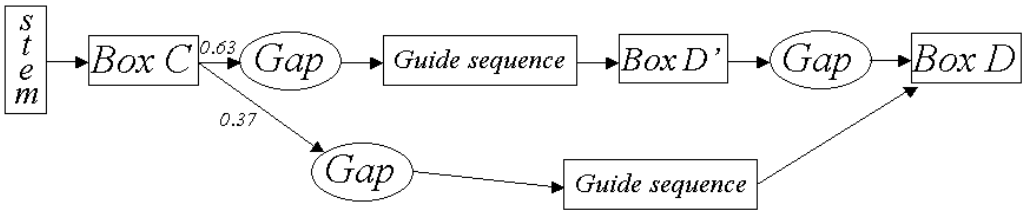
```
Descripteur rRNA 5s régionIII

parms
   wc += gu;

descr
   h5(len=2, seq="^AC$")
   h5(len=4, seq="^CYGN$")
   ss(tag='p1',
   seq="^YCCCATNCCGAAC$")
   h3
   ss(len=2)
   h3
```

5. Exemple semi probabiliste : SNOSCAN

SNOSCAN est un peu dans une classe à part dans le sens qu'il utilise les concepts de la recherche par contraintes comme pour les autres programmes de l'approche sûr mesure, mais utilise aussi l'approche probabiliste.

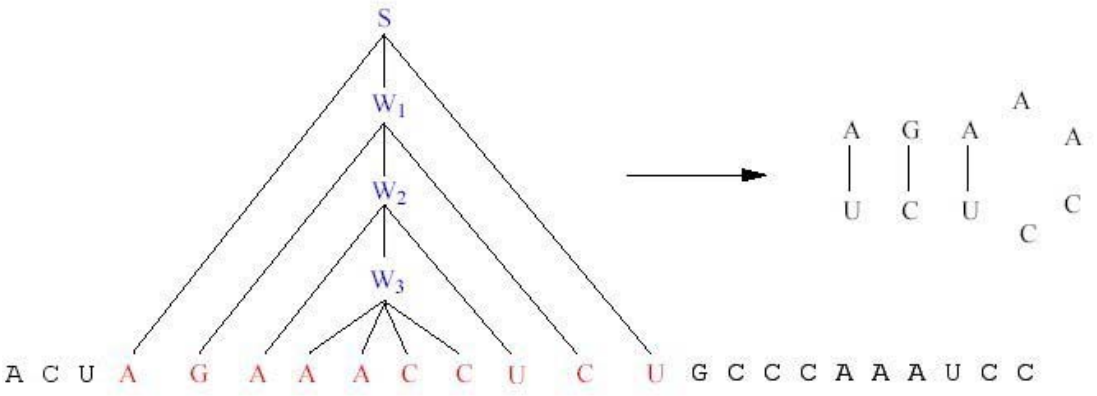


6. Problème de l'alignement à une grammaire hors contexte

Tel que vu précédemment, il est possible de représenter une structure secondaire de l'ARN à l'aide d'une grammaire. Considérer à titre d'exemple, la structure suivante :

$$
\begin{aligned}
S &\rightarrow A\,W_1\,U \mid C\,W_1\,G \mid G\,W_1\,C \mid U\,W_1\,A \\
W_1 &\rightarrow G\,W_2\,C \\
W_2 &\rightarrow A\,W_3\,U \\
W_3 &\rightarrow AACC \mid CACC \mid GACC \mid UACC
\end{aligned}
$$

Cette grammaire sert à produire un arbre de recherche qui sert à trouver tous les motifs définis par la grammaire dans une séquence :



Ceci démontre à quel point les grammaires peuvent être un outil puissant dans la recherche de structures secondaires. Par contre, peut importe la qualité d'un descripteur, la nature réussit toujours à créer des cas qui n'avaient pas été prévus. Ainsi, une grammaire pourrait être un outil encore plus puissant s'il était possible de la rechercher de façon approchée (avec les règles de la distance d'édition). Évidemment, ceci est possible grâce au monde merveilleux de la programmation dynamique !

<u>Algorithme de l'alignement approché d'une grammaire hors contexte</u> :

Pour toutes les règles A non terminales sous la forme pseudo normale de Chomsky ($A \rightarrow BC$, $A \rightarrow B$, $A \rightarrow a$, où B et C sont des non terminaux et *a* est un symbole terminal), $\delta(a, a_k)$, le coût d'aligner le symbole terminal *a* à a_k , le $k^{ième}$ caractère de S, gap(i,j), une fonction qui calcule le coût d'un gap sur l'intervalle i..j, $\sigma(a)$ le coût de laisser *a* non aligné, et $0 \leq i - 1 \leq j \leq n$:

$$
\begin{aligned}
G(A,i,j) = \min \{ \quad & \min_{A \rightarrow BC,\, k \in [i-1,\, j]} G(B,i,k) + G(C,k+1,j), \quad *** s[i..i-1] = \varepsilon \\
& \min_{A \rightarrow B} G(B, i, j), \\
& \min_{A \rightarrow a,\, k \in [i,\, j]} \delta(a, a_k) + gap(i,j) - \sigma(a_k), \\
& \min_{A \rightarrow a,\, k \in [i,\, j]} \sigma(a) + gap(i,j) \qquad \}
\end{aligned}
$$

7. Recherche de structures secondaires avec une grammaire

Maintenant, que nous avons un algorithme capable d'aligner une grammaire hors contexte avec une séquence, étendons le problème à des règles de grammaires capables de représenter les hélices biologiques.

Comme nous l'avons précédemment vu, une hélice peut être représentée par les règles suivantes :

$X \rightarrow n\ Y\ n'$, $X \rightarrow n\ Y$, $X \rightarrow Y\ n$, $X \rightarrow n$

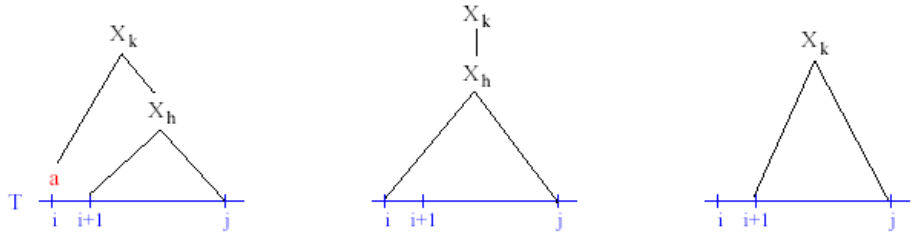
Étant donné une séquence d'ADN, maximisez le score de l'alignement de la grammaire !

- δ(a, b) : Score associé à l'alignement de *a* et *b*
- δ(a) : Score associé au indel de *a*
- p(a) : Score d'un appariement correct (*a* avec *a'*)

Exemple :

$$X_k \rightarrow a\ X_h$$

$(M_k)_{i,j}$: Max over all X_h of three values



a aligned with T[i] | Deletion of a | Deletion of T[i]

Algorithme :

$$G(X,i,j) = \max \{$$

$$\max_{X \rightarrow aY} \{\delta(a, a_i) + G(Y, i+1, j),\ \delta(a) + G(Y, i, j)\ \},\ \delta(a_i) + G(X, i+1, j),$$ | $X \rightarrow n\ Y$

$$\max_{X \rightarrow Ya} \{\delta(a, a_j) + G(Y, i, j-1),\ \delta(a) + G(Y, i, j)\ \},\ \delta(a_j) + G(X, i, j-1),$$ | $X \rightarrow Y\ n$

$$\max_{X \rightarrow aYa'} \{\delta(a, a_i) + \delta(a', a_j) + p(a) + G(Y, i+1, j-1),\ \delta(a, a_i) + \delta(a') + G(Y, i+1, j),\ \delta(a', a_j) + \delta(a) + G(Y, i, j-1),\ \delta(a) + \delta(a') + G(Y, i, j)\ \},\ \delta(a_i) + G(X, i+1, j),\ \delta(a_j) + G(X, i, j-1)$$ | $X \rightarrow n\ Y\ n'$

$$\}$$