Une petite faute

Remarque : La visite de suivi effectuée par l'auditeur/l'inspecteur responsable à la suite d'une non-conformité majeure doit avoir lieu le plus rapidement possible après la « date d'exécution de l'action corrective » indiquée à la partie B de la DAC (voir la section 6.3.3 du présent docment).

Lorsque l'auditeur/l'inspecteur responsable constate que l'action corrective a été exécutée et qu'elle est efficace, la DAC peut être classée.

(Source: Agence canadienne d'inspection des aliments. Manuel de mise oeuvre du Programme d'amélioration de la salubrité des aliments.)

«docment» : document, dorment, doucement, . . .

Distance entre deux mots

Distance d'édition : nombre d'opérations pour transformer un mot à un autre.

Opérations : sur caractères
- insertion (I *insert*)
- suppression (D *delete*)
- substitution (S *substitute*)
- identité (M *match*)

Calcul de distance

But : calculer le nombre *minimal* d'opérations pour la transformation.

Minimal : autant de M (identité) que possible.

Exemples : docment-document, docment-dorment, docment-doucement, docment-moments

On peut avoir plusieurs suites de la même taille minimale : abbc-axc

Calcul de distance 2

Idée : calculer les distances entre les préfixes successivement

Nos notions formelles :
- alphabet fini Σ (p.e. $\Sigma = \{\mathtt{a}, \ldots, \mathtt{z}\}$ ou $\Sigma = \{\mathtt{T}, \mathtt{G}, \mathtt{A}, \mathtt{C}\}$)
- mot ou séquence : une suite de caractères de Σ

Soit S une séquence.
- taille de S, denotée par $|S|$ = nombre de caractères
- caractère en position i : $S[i]$
- sous-mot $S[i..j]$: le mot formé par les caractères $S[i]$, $S[i+1]$,..., $S[j]$
- préfix : sous-mot de forme $S[1..i]$.

Calcul de distance 3

Def. Distance d'édition entre deux séquences S_1 et S_2 : nombre minimal de I, D, S dans une suite d'opérations qui transforme S_1 en S_2.

Thm. La distance d'édition est une fonction symétrique.
Preuve. I $\Leftrightarrow$ D.

Un fait intéressant :) la distance d'édition est parfois appellée distance de Levenshtein.

Calcul de distance 4

Def. Soit S et T deux séquences. On définit $D(i,j)$ par

$$D(i,j) = \begin{cases} \text{distance entre } S[1..i] \text{ et } T[1..j] & \text{si } i > 0 \text{ et } j > 0, \\ i & \text{si } j = 0 \\ j & \text{si } i = 0. \end{cases}$$

Donc $D(i,j)$ est la distance entre les deux préfixes de tailles i et j.

Thm. Si $i, j > 0$, on a

$$D(i,j) = \min\Big\{ D(i-1,j) + 1, \\ D(i,j-1) + 1, \\ D(i-1,j-1) + \mathbb{I}\{S[i] \neq T[j]\} \Big\}.$$

Calcul de distance 5

Preuve. 1. La dernière opération pour achever $D(i, j)$ doit être I, D, ou S/M.
2. Tous les trois sont possibles.

Donc on peut calculer $D(|S|, |T|)$ par récursion. . . pas une bonne idée. (La même valeur sera calculée plusieurs fois.)

Quand même on n'a que $(1 + |S|) \times (1 + |T|)$ appels récursifs possibles.

Au lieu d'explorer l'arbre de récursions en descendant, il est mieux de le faire de manière ascendante.

Tableau de calculs

Les cases contiennent les $D(i, j)$.

Parcours : ligne par ligne.

Exemple : $\texttt{AAAC} \rightarrow \texttt{AGC}$

Temps de calcul : $O(mn)$ (on remplit chaque case du tableau en un temps constant : trois comparaisons et deux ou trois additions)

Comment trouver une suite d'opérations qui correspond à $D(i, j)$? Enregistrer dans chaque case la direction du $\min$ de la recurrence.

Programmation dynamique

PD : récurrence+tableau+retrouver la solution optimale (en retraçant les pas à partir du case Sud-Est)

PD est utilisée pour resoudre des problèmes d'optimisation

1. sous-structures optimales ($\Rightarrow$ récurrence)
2. sous-problèmes superposés ($\Rightarrow$ tableau)

Mais pourquoi ça nous intéresse ?

L'alignement de séquences est une méthode utilisée très souvent en biologie moléculaire d'aujourd'hui.

Idée : similarité de séquences $\Rightarrow$ similarité de structures et fonctions (évolution de gènes/protéines : duplication+modification)

Régions conservées : importance pour la fonction/structure.

Exemple

protéine trypsine : souris (P07146 de SWISS-PROT) et grenouille (P70059 de SWISS-PROT)

```
souris      MSALLILALVGAAVAFPVDDDDKIVGGYTCRESSVPYQVSLNAGYHFCGGSLINDQWVVSAAHC
grenouille  MKFLVILVLLGAAVAFEDDD--KIVGGFTCAKNAVPYQVSLNAGYHFCGGSLINSQWVVSAAHC
```

Alignement des deux sequences : représente leur similarité.

Alphabet d'alignement : $\Sigma \cup \{-\}$.
Appariement : un couple de $\Sigma \cup \{-\}$ (p.e. $(\mathtt{C}, -)$, ou $(\mathtt{C}, \mathtt{G})$)
Alignement : suite d'appariements.

Relation entre opérations d'éditions et alignements (procédure et produit).

Alignement

Types d'appariements :
- (a, a) : occurence (*match*)
- (a, b) : erreur (*mismatch*)
- $(a, -)$ et $(-, a)$: espace

Score ou valeur d'appariements : mismatch et espace : -1 ; match : +1

Score ou valeur de l'alignement : somme des valeurs des appariements

Problème : trouver l'alignement avec le score maximal

Thm. Equivaut le problème de minimisation de la distance d'édition.

PD pour maximiser la similarité

Pondération

Pondération des opérations

Pondération de l’alphabet

Récurrences

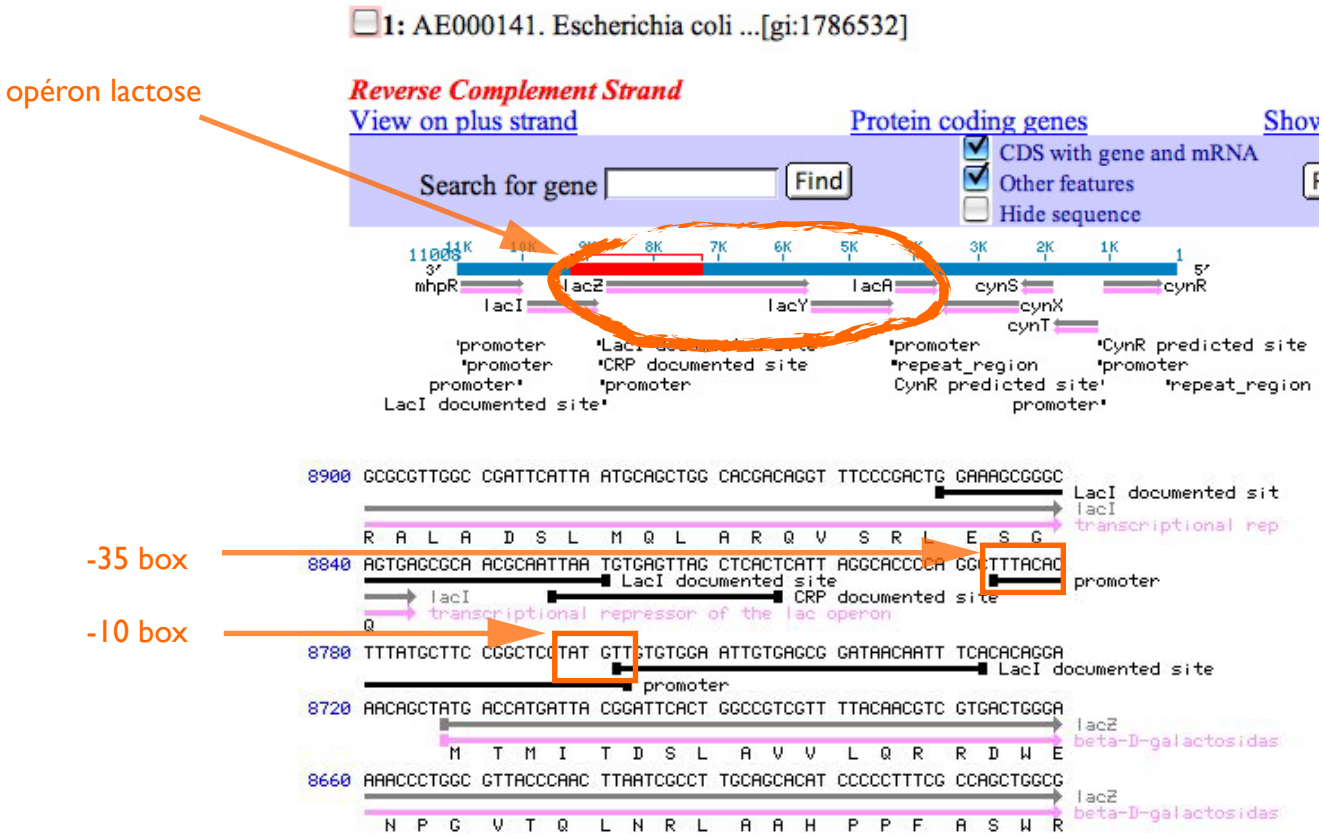
Alignement : variation 1

Trouver l'occurrence plus similaire d'une (courte) séquence dans une autre (longue)

Idée : calculer l'alignement pour les suffixes

Exemple 1 : séquences répétées (p.e. de type retrovirus)

Exemple 2 : promoteur sigma A : TATAAT...TTGACA



Alignement : variation 2

Trouver la région de similarité maximale entre deux séquences — alignement local

Exemple : domaines conservés (p.e. homeobox)

Les noms :
Needleman-Wunsch : problème de l'alignement global
Smith-Waterman : algorithme de l'alignement local

Alignement : variation 3

Exemple 1 : Alignement de ADNg et ADNc
(ADNc : ADN complémentaire pour ARNm transcrit)

Exemple 2 : pseudogènes

Trous

Pondérations :

- quelconque
- constante
- affine : $W = W_g + W_s \cdot$ longueur

Récurrences pour PD avec quatre fonctions

G - match/mismatch

E - trou dans S

F - trou dans T

$V = \max\{G, E, F\}$

À venir

Trouver le meilleur alignement dans une grande collection de séquences

Alignement de plusieurs séquences

Modèles de séquences : expressions régulières et chaînes de Markov