Notes de Bioinformatique

Philippe Gambette

14 juillet 2005 - version 3.1

Table des matières

1 Avant-propos **7**

I Analyse de séquences **9**

2 Comparaison de séquences deux à deux **13**
2.1 Alignement global optimal 13
2.1.1 Score élémentaire 13
2.1.2 Matrice de score 14
2.1.3 Parcours de la matrice 14
2.2 Alignement local optimal 14
2.3 Matrices de score 16
2.3.1 Matrices de distance entre séquences nucléiques 16
2.3.2 Matrices protéiques 16

3 Comparaison d'une séquence à une banque **19**
3.1 FASTA 19
3.2 BLAST 19

4 Alignement multiple **21**

5 Motifs et modèles **23**
5.1 Motifs 23
5.2 Profils 23
5.3 Modèles de Markov cachés 24

6 Le séquençage des génomes **25**
6.1 Le génome 25
6.2 Le décryptage du génome 26
6.2.1 L'approche historique 26
6.2.2 Les méthodes de reconstruction du génome 28

7 Les microARN **29**

II Phylogénie **31**

8 Les arbres phylogénétiques **35**
8.1 Optimisation de distance 36

8.1.1 UPGMA . . . 36
8.1.2 Neighbour-joining . . . 37
8.1.3 Analyses de saturation . . . 37
8.2 Optimisation de la vraisemblance . . . 37
8.3 Optimisation de la parcimonie . . . 38
8.4 Enracinement d'un arbre phylogénétique . . . 38
8.5 Améliorations par réarrangements topologiques . . . 38
8.6 Optimisation : du local au global . . . 39
8.6.1 Le recuit simulé . . . 39
8.6.2 Le grand déluge . . . 39
8.7 Les arbres de duplication . . . 39

9 Le cas du VIH **43**
9.1 Les origines du VIH . . . 43
9.2 Les raisons de la difficulté de mise au point d'un vaccin . . . 45
9.3 VIH et génétique . . . 45
9.4 En conclusion . . . 46

10 Les réseaux phylogénétiques **47**
10.1 Le consensus d'arbres phylogénétiques . . . 47
10.2 La visualisation des conflits phylogénétiques par les splits graphs 47
10.3 La recherche des recombinaisons . . . 48
10.4 La recherche des transferts horizontaux . . . 48

III Modélisation, reconstruction et analyse de réseaux biologiques **49**

11 Introduction aux réseaux biologiques **51**
11.1 Le contexte de ce thème . . . 51
11.2 L'introduction des réseaux . . . 52

12 La cellule **55**
12.1 Les caractéristiques de base de la vie . . . 55
12.2 Les raisons de l'intérêt pour les cellules . . . 56

13 Les bases de la reconstruction de réseaux **57**
13.1 Les puces à ADN . . . 57
13.2 Les analyses de bas niveau . . . 58
13.2.1 Analyse d'image . . . 58
13.2.2 Normalisation . . . 58
13.3 Clustering . . . 59
13.4 Le biclustering . . . 60
13.5 La reconstruction de réseaux de transcription . . . 60

14 Calcul de membrane **61**
14.1 Notations . . . 61
14.2 Syntaxe . . . 62
14.2.1 Actions . . . 62
14.2.2 Branes . . . 62
14.2.3 Processus . . . 62

14.2.4 Règles 62
14.2.5 Versions informelle et bitonale des règles 62
14.3 Equivalence projective 62

IV Annexe **65**

15 Mise en pratique sous Linux **67**
15.1 Commandes de base 67
15.1.1 Commandes de gestion des dossiers 67
15.1.2 Commandes de gestion des processus 67
15.1.3 Types de transferts de fichiers 68
15.1.4 Transfert FTP shell 68
15.1.5 Compression 68
15.1.6 Permissions 68
15.1.7 Commandes de gestion des fichiers texte 68
15.1.8 Outils de base 69
15.2 Centres de recherche en bio-informatique 70

16 Lexique anglais/français **71**

Chapitre 1

Avant-propos

Ce qui suit provient de notes prises au cours de ma scolarité en rapport avec la bioinformatique. La première version, et donc les fondamentaux de ces notes, à propos de l'alignement de séquences et de la phylogénie, sont issus du cours de bioinformatique de Gilles Pitiot du second semestre de l'année 2003/2004, proposé aux élèves de première année des départements informatique et mathématique de l'ENS Cachan.

J'ai ensuite ajouté quelques notes prises pendant les cours de M2 donné au MPRI (Master Parisien de Recherche en Informatique) par Vincent Schächter, Vincent Danos et François Fages.

Plusieurs conférences de biologie ou de bioinformatique ont aussi servi à compléter ce document, une passionnante à propos de la variabilité du VIH et des conséquences sur son traitement donnée par Simon Wain-Hobson à l'ENS Cachan le 11 mai 2004, et celles du cycle Mathematics of Evolution and Phylogeny 2005 à l'Institut Poincaré, organisé par Olivier Gascuel.

Mes stages de recherche m'ont aussi permis d'ajouter des éléments ou de corriger certains paragraphes. Celui de fin de licence au LIRMM[1] avec Olivier Gascuel et Denis Bertrand, qui m'ont permis d'assurer certaines connaissances. Et celui de fin de première année de master avec Daniel Huson et son équipe du ZBIT[2], qui m'ont donnée une vue plus large, plus détaillée. En particulier, moult zones d'ombres ont été clarifiées par le cours de Daniel Huson de Bioinformatique 2 à l'Université de Tübingen. L'intervention de Tobias Dezulian a permis de découvrir les microARN7.

La bioinformatique est un domaine qui utilise beaucoup de termes anglais, c'est pourquoi vous en trouverez beaucoup dans ces notes. Un lexique en annexe indique l'équivalence en français en cas de besoin.

J'ai tenté de produire moi-même le maximum d'illustrations. Si toutefois vous détectez un problème de copyrights, merci de m'en faire part par mail à `p.gambette@free.fr`.

Et évidemment, prenez tout le contenu de ce document avec une extrême précaution et un sens critique aigu. Non seulement je vous invite à douter perpétuellement de ce que vous y lisez, mais en plus me signaler vos doutes, mes erreurs, ou encore les liens morts, à l'adresse sus-citée. D'ailleurs, si vous vous

[1] Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, http://www.lirmm.fr/xml/fr/lirmm.html

[2] Zenter für Bioinformatik Tübingen

sentez d'attaque, après la lecture de ce document et d'autres sources pour en vérifier les informations, vous pourrez vous lancer dans une contribution à l'encyclopédie en ligne modifiable par ses visiteurs Wikipedia[3].

En ce qui concerne la bibliographie, vous pourrez retrouver un certain nombre de références en utilisant Google ou Google Scholar[4].

Enfin, ce document est soumis à la license Creative Commons "by-nc" : paternité, pas d'utilisation commerciale, modifications autorisées.[5]

[3] http://fr.wikipedia.org
[4] http://scholar.google.com
[5] http://creativecommons.org/licenses/by-nc/2.5/

Première partie

Analyse de séquences

Pour découvrir cette partie en rapport avec l'analyse de séquences, vous pourrez vous référer au cours disponible sur le site *infobiogen* [10], très accessible et pédagogique. Vous pourrez aussi aller voir les notes d'algorithmique et bioinformatique de Vincent Zoonekynd [43], qui sont très riches, avec de nombreux algorithmes détaillés voire implémentés, et une jolie bibliographie. De nombreuses références se trouvent aussi dans le cours de bioinformatique de Paris V sur l'analyse et les banques de séquences[6].

La compréhension détaillée des algorithmes cités sera permise grâce à la bibliographie qui recense les articles fondateurs pour chacun des problèmes abordés.

[6] http ://www.dsi.univ-paris5.fr/bio2/biocours/intro.html

Chapitre 2

Comparaison de séquences deux à deux

Cette partie de la bioinformatique consiste notamment à identifier des séquences de nucléotides communes chez l'homme et d'autres animaux, afin d'identifier des gènes communs ou de construire des arbres phylogénétiques.

Afin de réaliser cette analyse, on doit prendre garde à la complexité des algorithmes utilisés. En effet, les bases de données actuelles contiennent jusqu'à 700000 protéines. Les fichiers de séquences des génomes humains font jusqu'à 5 Go. Il est donc la plupart du temps indispensable de trouver des algorithmes de complexité linéaire.

Nous allons découvrir que les algorithmes d'alignement se fondent bien souvent sur la notion de *scores*, qui indiquent la validité de l'alignement réalisé.

2.1 Alignement global optimal

On désire comparer les séquences communes à deux mots A et B (formés des lettres `A`,`C`,`G`,`T`). Nous effectuerons cette comparaison grâce à l'algorithme de Needleman Wunsch datant de 1969 [37] qui est en complexité cubique. Un algorithme un peu plus joli en complexité linéaire est présenté dans l'*Introduction à l'Algoritmhique* [12].

2.1.1 Score élémentaire

On construit un tableau à double entrée (une matrice M, dite "matrice de scores élémentaires"), en plaçant A horizontalement, et B verticalement. On remplit alors le tableau (figure 2.1) avec :

- $M(i,j) = 1$ si $A[i] = B[j]$.
- $M(i,j) = 0$ sinon.

On peut déjà remarquer que les diagonales haut/gauche-bas/droit de 1 indiquent des séquences communes à A et B. Ce *dot-plot* (graphe par matrice de points) permet donc de faire un premier repérage visuel des séquences.

	A	A	C	A	T	G	G	T	A	C	A	C	C
A	1	1	0	1	0	0	0	0	1	0	1	0	0
T	0	0	0	0	1	0	0	1	0	0	0	0	0
A	1	1	0	1	0	0	0	0	1	0	1	0	0
C	0	0	1	0	0	0	0	0	0	1	0	1	1
A	1	1	0	1	0	0	0	0	1	0	1	0	0
T	0	0	0	0	1	0	0	1	0	0	0	0	0
G	0	0	0	0	0	1	1	0	0	0	0	0	0
T	0	0	0	0	1	0	0	1	0	0	0	0	0
A	1	1	0	1	0	0	0	0	1	0	1	0	0
G	0	0	0	0	0	1	1	0	0	0	0	0	0
G	0	0	0	0	0	1	1	0	0	0	0	0	0
C	0	0	1	0	0	0	0	0	0	1	0	1	1
C	0	0	1	0	0	0	0	0	0	1	0	1	1

FIG. 2.1 – Exemple de matrice de dot-plot

2.1.2 Matrice de score

Nous pouvons déterminer la séquence optimale par l'algorithme ci-dessous.

On parcourt la dernière ligne, puis la dernière colonne de la matrice, puis ce qui reste à parcourir de l'avant-dernière ligne, et de l'avant-dernière colonne... en réalisant l'opération suivante à chaque case $M(i,j)$ de la matrice :

$$M(i,j) = M(i,j) + \max_{\substack{k\in[j+1,n]\\ l\in[i+1,m]}} \{M(i+1,k)\} \cup \{M(l,j+1)\}$$

C'est cette étape de calcul de $M(i,j)$ qui est effectué en temps linéaire ici, et en temps constant dans [12].

On obtient alors le tableau de la figure 2.2

2.1.3 Parcours de la matrice

On effectue alors un parcours du graphe en partant de la case maximale parmi celles de la première ligne et de la première colonne, et en choisissant à chaque fois la case adjacente maximale dans le coin en bas à droite, ou celle en bas, ou celle à droite, jusqu'à ce que l'on atteigne la dernière ligne, ou la dernière colonne (Figure 2.3).

Enfin, on parcourt ce chemin en partant de la case (0,0) : à chaque fois que le numéro de la case parcourue augmente, on est sur une lettre commune aux deux séquences.

2.2 Alignement local optimal

Le but est à présent de trouver une région de forte homogénéité. On va donc privilégier les alignements locaux, et donc empêcher le non-aligné en affectant des poids négatifs en cas de trous ou de misappariements. Ce sont Smith et Waterman qui proposent un algorithme en 1980 [40]. Après avoir réalisé la

	A	A	C	A	T	G	G	T	A	C	A	C	C
A	10	9	7	7	6	6	6	5	5	3	2	1	0
T	9	8	7	6	6	5	5	6	4	3	2	1	0
A	9	9	7	7	5	5	4	4	5	3	2	1	0
C	7	7	8	6	5	5	4	4	3	4	2	2	1
A	7	7	6	7	5	5	4	3	4	3	3	1	0
T	5	5	5	5	6	5	4	4	3	3	2	1	0
G	5	5	5	5	4	5	5	3	3	3	2	1	0
T	5	5	5	4	5	3	3	4	3	3	2	1	0
A	5	5	4	5	4	3	2	2	3	2	3	1	0
G	3	3	3	3	3	4	3	2	2	2	2	1	0
G	2	2	2	2	2	3	3	2	2	2	2	1	0
C	1	1	2	1	1	1	1	1	1	2	1	2	1
C	0	0	1	0	0	0	0	0	0	1	0	1	1

FIG. 2.2 – Exemple de matrice obtenue après la deuxième partie de l'algorithme de Needleman Wunsch

	A	A	C	A	T	G	G	T	A	C	A	C	C
A	10												
T	9												
A		9											
C			8										
A				7									
T					6	5							
G							5						
T								4					
A									3				
G										2			
G											2		
C												2	
C													1

FIG. 2.3 – Exemple de chemin obtenu après la troisième partie de l'algorithme de Needleman Wunsch

matrice de score élémentaire, que l'on appellera H, on complète la matrice de score avec la formule suivante :

$$M(i,j) = \max\{(H(i,j)+M(i-1,j-1)), \max_{k\geq 1}\{H(i-k,j)-W_k\}, \max_{l\geq 1}\{H(i,j-l)-W_l\}, 0\}$$

avec par exemple :

$$W_k = 1 + \frac{1}{3}k$$

La matrice obtenue permet de repérer les alignements locaux. Et l'algorithme se fait en temps cubique sur la taille des 2 séquences : quadratique pour visiter toutes les cases de la matrice, et linéaire pour calculer $M(i,j)$.

2.3 Matrices de score

Lors des calculs précédents de matrice de score élémentaire, on considérait que les similarités, ou les différences, avaient toutes la même importance, en leur affectant un score de 1 ou 0 respectivement. En réalité, il faut prendre en compte les ressemblaces entre deux lettres différentes, par exemple entre `C` et `U`, puisqu'il suffit d'une oxydation pour passer de la cytosine à l'uracile.

2.3.1 Matrices de distance entre séquences nucléiques

Quelques définitions :

- distance : mesure du nombre de changements évolutifs entre deux génomes depuis leur divergence à partir d'un ancêtre commun.
- score élémentaire : entre deux acides aminés ou entre deux nucléotides.
- score global : somme des scores élémentaires sur une séquence.

On utilisait précédemment comme matrice de distance entre les nucléotides la matrice unitaire :

	A	C	G	T
A	1	0	0	0
C	0	1	0	0
G	0	0	1	0
T	0	0	0	1

En fait, le passage des nucléotides `A` à `G`, `G` à `A`, `C` à `T`, et `T` à `C` se fait par transition. Les autres passages entre nucléotides se font par transversion, et sont plus rares. Il faut donc utiliser une matrice de distance qui tient compte de ces différences, la *matrice de transition/transversion* :

	A	C	G	T
A	$1,36$	$-1,6$	$-0,37$	$-1,6$
C	$-1,6$	$1,36$	$-1,6$	$-0,37$
G	$-0,37$	$-1,6$	$1,36$	$-1,6$
T	$-1,6$	$-0,37$	$-1,6$	$1,36$

2.3.2 Matrices protéiques

Matrice unitaire :

On ne s'en sert pas : il est en effet nécessaire de prendre en compte les ressemblances entre acides aminés. Ces ressemblances sont de plusieurs types :

- de même charge
- hydrophiles / hydrophobes
- type de groupement latéral : polaire (groupe I), polaire non chargé (groupe II), chargé (groupe III)

Matrice de changement minimum / de code génétique :

On regarde sur le code génétique le nombre minimal de transformations sur un codon pour passer d'un acide aminé à l'autre. En raison de la redondance

du code génétique, le nombre de transformations est en effet variable : il peut être égal à 1 ou 2 pour la transformation phénylalaline / sérine par exemple.

Remarque : on utilisera par la suite le code des acides aminés : `A` pour `Ala` (alamine), `C` pour `Cys` (cytosine).

Matrices liées aux propriétés chimiques :

Par exemple, la matrice de Johnson et Overington (1993) se fonde sur les analogies de structure tridimensionnelle.

Matrices liées à l'évolution :

La plus utilisée est PAM[1] (Point Accepted Matrix), proposée par Margaret Dayhoff en 1978. Elle provient d'un alignement à la main de 3000 protéines d'une même famille afin de déterminer les mutations les plus fréquentes d'un acide aminé à un autre. La matrice obtenue, PAM1, (respectivement PAM250), est construite en considérant que la probabilité d'une mutation est de 1 (resp. 250) pour 1000 acides aminés.

$$\mathrm{PAM}_i^j = P(\mathrm{A.A.}_i \rightarrow \mathrm{A.A.}_j)$$

Si PAM_i^i est élevé, cela signifie que l'acide aminé $\mathrm{A.A.}_i$ est très souvent conservé au cours de l'évolution.

Cette matrice PAM présente toutefois deux limites. Tout d'abord, l'alignement initial a été fait sur la totalité de la séquence, incluant les régions non conservées. De plus, il a été réalisé sur des séquences de protéines connues dans les années 70, des protéines globulaires, et ceci a peut-être biaisé les probabilités de mutation.

Les matrices BLOSUM proposées par Henikoff permettent d'éviter le premier inconvénient. puisque ce sont seulement les parties communes sans trous des protéines (et donc à forte similarité) qui ont été analysées. De plus, des protéines plus récentes et plus diverses ont été utilisées, ce qui a montré que les résultats sur les protéines globulaires n'étaient pas vraiment biaisés.

BLOSUM est devenue une matrice de référence, plus que PAM, notamment BLOSUM62 utilisée avant l'apparition du programme BLAST.

Finalement, il faut tout de même garder à l'esprit qu'il faut choisir la matrice de score élémentaire en fonction de nos objectifs. En effet, si l'on recherche des séquences très similaires (plus de 80%), il sera préférable d'utiliser PAM1 ou BLOSUM80, BLOSUM 45 ou PAM250 pour des séquences moins similaires (30%), et BLOSUM62 ou PAM120 pour un juste milieu. Et en dessous de 20% de similarité, on ne peut pas vraiment être sûr d'avoir aligné deux séquences.

Mais ces algorithmes, s'ils sont optimaux, n'en sont pas moins très lourds, et trop lents pour réaliser des comparaisons de séquences avec une base de données. Pour ce faire, il nous faut des algorithmes qui évitent de réaliser quelques alignements inutiles.

[1] http://www.esil.univ-mrs.fr/%7Edgaut/Cours/pam.html, lien vérifié le 28/06/2005

Chapitre 3

Comparaison d'une séquence à une banque

3.1 FASTA

Cet algorithme [39], proposé par Pearson et Lipman en 1988 se déroule en quatre étapes, et se fonde sur une recherche de mots communs (*ktup*) entre la séquence à analyser et toutes celles de la base. Le k de *ktup* est la taille des mots à comparer entre les deux séquences, c'est un paramètre que l'on peut choisir entre 1 et 6 (2 pour les protéines, 4 à 6 pour l'ADN).

Pour chaque séquence de la base de données, on la compare à la séquence à analyser en effectuant les 4 étapes suivantes :

1- On effectue une sorte de dot-plot sur plusieurs lettres : on affecte 1 à la case correspondant à la première lettre de tout *ktup* commmun à deux séquences. Bien entendu, plus k est grand, plus on est sélectif.

2- Puis on calcule les scores sur chaque diagonale de la matrice en mettant une pénalité en cas de misappariement et donc en ne prenant pas en compte les délétions et les insertions, puis on garde les 10 diagonales de score maximal.

3- On utilise alors une matrice de score (PAM250) pour calculer le score sur les dix meilleures diagonales, entre le premier et le dernier appariement, et en utilisant éventuellement un seuil. On obtient un score *init 1*.

4- On procède alors au rattachement des diagonales en affectant un score de jonction pour chacune des portions de diagonales, et en autorisant donc les délétions/insertions. On obtient donc un score *init n*. Ce sont les *init i* qui seront utilisés pour classer les séquences de la base.

5- On peut optimiser en utilisant les algorithmes de Needleman Wunsch ou Smith Waterman, mais seulement sur la zone contenant les dix diagonales.

Cet algorithme a toutefois un inconvénient : à la deuxième étape, on a ajouté des régions de faible homologie.

3.2 BLAST

Contrairement à FASTA, on ne sélectionne ici que les séquences qui contiennent des régions de forte homologie, en quatre étapes :

1- Pour chaque séquence, on crée un tableau des sous-mots de cette séquence d'une longueur w fixe, que l'on appelle les *graines*.

2- On recherche[1] alors chaque mot dans toute séquence de la base, et on réalise donc un alignement de longueur w (3 en général).

3- En partant de la graine, on étend l'alignement, tant que le score de cet alignement reste supérieur à un seuil S, pour obtenir un *MSP* (*Maximum Segment Pair*).

4- L'alignement ayant le plus grand score dans la séquence est appelé HSP. On va donc pouvoir comparer les séquences de la base à la séquence de référence en classant ces séquences en fonction de leur HSP.

Toutefois, cet algorithme [3] présente une limite : si w=10, et qu'une séquence est similaire à 90% par rapport à la séquence de référence, avec un misappariement tous les 10 acides aminés, elle ne sera pas repérée. BLAST2 règle ce problème, en prenant en compte les délétions à la troisième étape en ajoutant des pénalités en cas d'ouverture de gap, et même en cas d'extension de gap. BLAST2 est de plus trois fois plus rapide que son aîné.

[1] Cette expression anodine, "on recherche" pose un énorme problème : comment rechercher efficacement un mot, dans ce cas précis, assez court, dans une base de données de plusieurs gigaoctets ? Toute l'astuce consiste ici à utiliser des arbres suffixe ou tries (prononcer "trailles") pour stocker les arbres de données. Cette structure bien connue en informatique permet de rechercher un mot dans un autre en temps la longueur du mot que l'on recherche. Formidable ? Oui, mais il faut d'abord construire l'arbre suffixe de la manière la plus rapide (en temps linéaire), et/ou la plus compacte possible. Pour cela, on pourra se référer à [28], ou chercher des informations sur l'algorithme d'Ukkonen.

Chapitre 4

Alignement multiple

L'alignement multiple permet en particulier de repérer des séquences fortement conservées dans des protéines d'une même famille, qui ont donc une importance particulière dans la fonction de celles-ci. L'algorithme basique consistant à essayer tous les alignements étant bien sûr exponentiel, CLUSTAL [24] propose une méthode astucieuse en trois étapes :

1- On effectue un alignement de toutes les séquences deux à deux, ce qui permet d'obtenir un score pour chaque paire, et donc une matrice des scores.

2- En utilisant cette matrice des scores, on construit un dendrogramme : un arbre qui organise les séquences en fonction de leurs mutations par rapport à la séquence originelle supposée, comme illustré dans la figure **??**. Le dendrogramme est lié au concept de phylogénie que nous développerons en partie II, et c'est donc un algorithme basique de phylogénie, UPGMA (voir section 8.1.1), qui est utilisé pour construire ce dendrogramme.

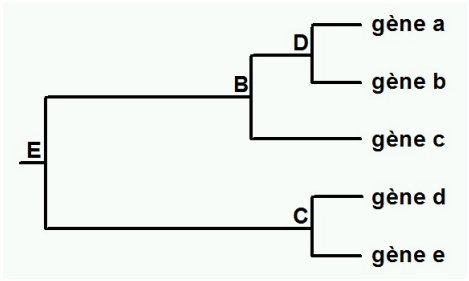


FIG. 4.1 – Un dendrogramme sur des gènes de différentes espèces (des homologues et des paralogues), dont les noeuds indiquent donc les spéciations ou les duplications de gènes.

3- On itère le processus suivant : on aligne les deux séquences les plus proches, deux feuilles-soeurs du dendrogramme de profondeur maximale, (en utilisant 4 scores pour acides aminés différents : 0 si misappariement avec deux acides aminés "très différents", 1 si misappariement avec deux acides aminés "proches", 2 si consensus sur un acide aminé peu significatif, 3 si consensus sur un acide aminé significatif : `Cys`, `Phe`, `Trp`, `Tyr`) puis on en effectue

un consensus, en mettant des `X` aux positions des misappariements, et on réinjecte ce consensus dans le dendrogramme.

Chacune de ces étapes est effectuée de façon indépendante, et des résultats intermédiaires sont disponibles à la fin de chaque étape. Le dendrogramme, notamment, a un intérêt certain : on peut y identifier les spéciations et les duplications de gènes. En effet, les protéines similaires peuvent l'être de deux manières : les *orthologues* sont des protéines similaires chez deux animaux d'espèce différente, qui ont dérivé de leur ancêtre commun par une spéciation, alors que les *paralogues* sont des protéines similaires au sein d'une même espèce, qui ont donc dérivé suite à une duplication au sein du gêne de leur ancêtre commun.

La deuxième étape de l'algorithme CLUSTAL présente toutefois un point faible, détaillé en section 8.1.1, qui incite à envisager s'utiliser un algorithme plus robuste pour construire le dendrogramme.

Chapitre 5

Motifs et modèles

5.1 Motifs

L'alignement multiple abordé précédemment permet de déduire des arbres phylogénétiques, comme nous le verrons au chapitre 8, et aussi, ce qui nous intéresse à présent, des consensus qui permettent d'identifier des zones fortement conservées. Utiliser ces consensus, que l'on appelle *motifs*, pourrait nous permettre de repérer des protéines mises de côté jusque là. La détection de motifs est aussi motivée par l'étude des facteurs de transcription, voir section 6.1. On a donc créé des banques de *motifs*, comme PROSITE, qui contiennent des motifs de 3, 4 ou 5 acides aminés.

Ceci permet d'identifier certains sites dans une protéine à étudier. Un site de glycolisation, par exemple, permet d'attacher des glucides sur une chaîne peptidique, sur l'asparagine souvent. En fait, dans ce type de site, une enzyme reconnaît des acides aminés et y attache ses glucides. Ceci nous montre que le critère de découverte du motif n'est pas suffisant pour assurer que la protéine possède un tel site : si le motif est "caché" à l'intérieur de la protéine par exemple, il ne sera pas reconnu par l'enzyme, et donc inutile pour la protéine.

5.2 Profils

Le *profil* se fonde de façon plus directe sur la notion de consensus, en gardant toutes les possibilités d'acides aminés présentes dans l'alignement multiple trouvé précédemment. On peut donc stocker ces profils dans des banques (PROSITE, PRINTS, PFAM). En outre, on peut remarquer que l'on trouve dans ces profils des acides aminés à plusieurs endroits avec un score différent. Il faudrait donc trouver un modèle permettant d'associer un score aux nucléotides en tenant compte de leur position.

C'est ce que fait PSI-BLAST [4], en utilisant la méthode suivante :

1- on commence par un BLAST.

2- on fait un alignement multiple.

3- on calcule une nouvelle matrice de comparaison qui dépend de la position.

4- on refait un BLAST en utilisant cette matrice, et on repart à l'étape 2.

Avec cet algorithme, on trouve donc les protéines par paliers au cours des multiples itérations : une famille est identifiée, puis on repère par hasard une protéine éloignée, donc on identifie toute sa famille, et ainsi de suite... PSI-BLAST a été utilisé pour construire la base de données IMPALA.

5.3 Modèles de Markov cachés

Le modèle de Markov caché (*Hidden Markov Model* ou *HMM*) est un outil performant pour reconnaître par exemple des familles de gènes. C'est un automate : un ensemble d'états reliés par des transitions, étiquetées par une certaine probabilité, et qui émettent des nucléotides ou acides aminés. Ainsi, pour chaque famille de gènes/protéines, on aura un modèle de Markov caché associé. On en comprend bien l'intérêt en bio-informatique depuis une dizaine d'années. Le fonctionnement des modèles de Markov est expliqué très clairement dans [15] et de façon plus approfondie dans [26]. HMMER est le programme qui permet de travailler avec ces modèles en bioinformatique.

Cet outil est donc l'un de ceux disponibles pour l'apprentissage automatique en général. On peut par exemple l'utiliser dans le domaine de la reconnaissance vocale : là, on a un modèle de Markov associé à un individu si l'on cherche à transcrire ses paroles, ou encore un modèle de Markov associé à chaque instrument si notre but est de différentier plusieurs instruments de musique. D'autres outils d'apprentissage automatique sont fréquemment utilisés en bioinformatique : le calcul bayésien ou les machines à vecteurs de support.

A faire : dessin d'un HMM, et algorithmes de Viterbi et Baum-Welch (EM) ...

Chapitre 6

Le séquençage des génomes

6.1 Le génome

Le génome est l'ensemble des gènes (gène : unité transcriptionnelle qui code pour une protéine, pouvant être responsable d'un caractère phénotypique, constitué de parties codantes, les exons, et d'introns, cf figure 6.1) d'un individu. Celui des procaryotes (cellules sans noyau, pro : avant) ne contient pas d'introns et a une grande densité de gènes, alors que celui des eucaryotes (animaux, végétaux, protozoaires... le préfixe *eu* signifiant *vrai*) ont des gènes à introns qui sont parfois morcelés sur la chaîne d'ADN. Le génome humain contient 3 Giga paires de bases (Gbp), ce qui représente 750 Mo de données. Parmi ce génome, 1 Gbp ne sont pas répétitives, 320 Mbp sont inclues dans des gènes, et 160 Mbp seulement sont codantes.

Il reste encore quelques zones d'ombre dans nos connaissances actuelles sur le génome des eucaryotes, notamment sur le rôle des télomères (les terminaisons physiques des chromosomes, souvent constitués de la séquence `TTAGGG` répétée). Par exemple, ce n'est qu'en 2003 qu'une équipe tchèque a découvert que contrairement à l'idée reçue que les télomères de tous les végétaux étaient constitués de la séquence `TTTAGGG`, certains avaient en fait la séquence `TTAGGG` [16], et l'oignon une autre séquence encore indéterminée en 2005...

On émet aussi plusieurs hypothèses sur le rôle des régions intergéniques, parfois appelées à tort *ADN poubelle* ou *bulk ADN*. Ces régions, qui contiennent les SINE (10% du génome humain) et les LINE (15% du génome humain) ne respectent pas la phylogénie, et sont très variables entre les espèces. Sont-elles des traces d'anciens gènes, de nouveaux gènes, ou permettent-elles l'attachement de l'ADN à la membrane ?

Parmi les autres séquences répétées, on distingue celles qui sont répétées en tandem, issues de recombinaisons inégales, et dont l'histoire peut être représentée par les arbres de duplications abordés en section 8.7.

On s'interroge aussi sur le *répresseur* (silencer) ou l'*activateur* (enhancer), qui arrivent à moduler l'activation du promoteur, et donc la transcription, bien qu'ils soient éloignés du gène dans la chaîne d'ADN. Ces activateurs ou répresseurs sont eux-mêmes régulés par des protéines TF (Transcription Factor)[1], qui

[1] voir les jolis schémas de http://www.csu.edu.au/faculty/health/biomed/subjects/molbol/gene.htm

se sont réparties dans les cellules dès les premières divisions de l'ovule [1]. Chaque gène est précédé d'une *région de régulation* (regulatory region) qui s'étend sur 100 à 1000 paires de bases. Dans cette région se trouvent les sites de fixation aux facteurs de transcription, qui peuvent donc s'y attacher pour réguler la transcription. Ces sites de fixation sont courts, apparaissent à plusieurs endroits sur la séquence d'ADN, avec éventuellement quelques différences. C'est une des motivations de la recherche de motifs, abordée en section 5.1.

La régulation par les microARN, abordée en section 7 est aussi une découverte très récente (abondance de publications sur le sujet en 2002), qui montre bien le rôle primordial que peuvent jouer certains introns.

On connaît toutefois l'aspect général du génome, et son devenir lors de la transcription, c'est ce qui est résumé dans la figure 6.1.

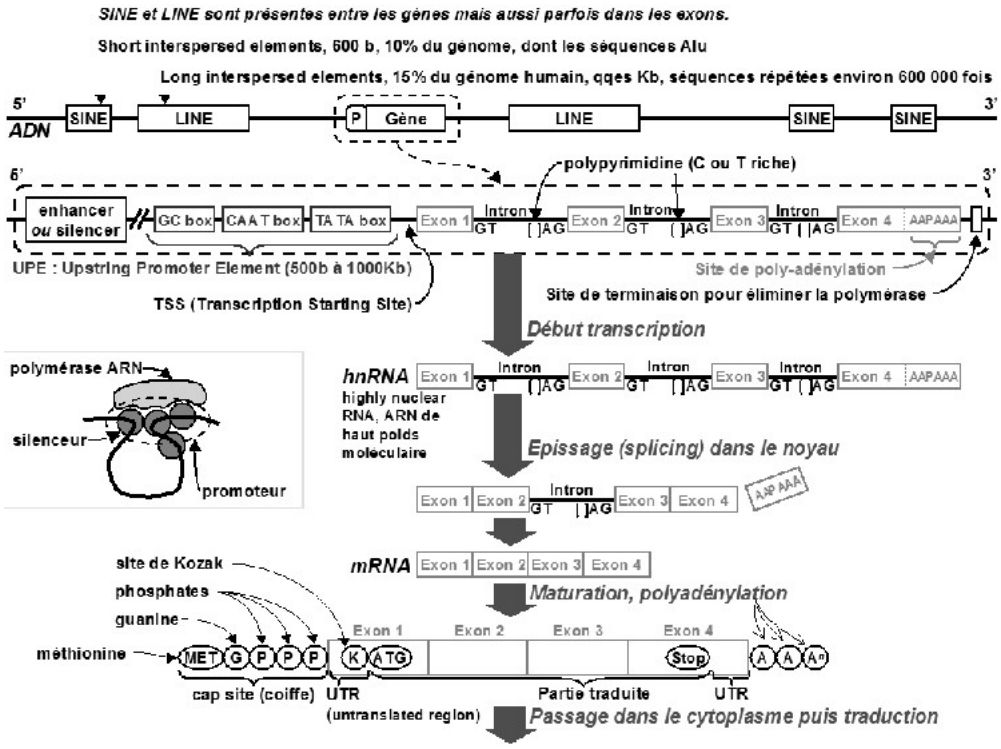


FIG. 6.1 – Le génome et la transcription chez les eucaryotes

6.2 Le décryptage du génome

6.2.1 L'approche historique

Cette approche est aussi appelée *génétique inverse* ou *clonage positionnel*, et consiste à se fonder sur le phénotype pour identifier l'endroit du génome où est codé le caractère.

Pour ce faire, on croise deux lignées pures avec deux caractères différents. Si les deux caractères se transmettent souvent en même temps, les deux gènes sont

proches sur la chaîne d'ADN. Pour être plus précis, on peut mettre des marqueurs sur l'ADN et regarder leur transmission en parallèle avec la transmission des caractères pour identifier à peu près l'emplacement du gène, et ainsi faire une carte du génome.

Evidemment, cette méthode est fondée sur un nombre élevé de recombinaisons. On peut donc l'appliquer facilement sur des petits pois comme l'avait fait Mendel (1865) qui a deviné le concept de gènes indépendants et liés. En revanche, c'est plus difficile chez l'homme. En effet, les distances identifiables sont alors de 1 cM (morgan : unité choisie en hommage à Thomas H. Morgan, fondateur de la génétique, qui indique la probabilité que deux gènes soient séparés en une génération , 1cM : probabilité de 1%), soit un million de paires de bases (l'ADN humain contient 3.10^9 paires de bases).

Supposons donc à présent que l'on a réussi à identifier une région de 500 Kb (qui peut contenir une dizaine de gènes) où pourrait se trouver le gène responsable d'une maladie. A l'époque, on ne savait pas isoler les gènes ni les purifier, et on ne disposait d'aucune séquence. Mais dans les années 70, on a réussi à cloner de gros fragments d'ADN : on injecte ce gros fragment dans une espèce qui le multiplie massivement, une bactérie par exemple (figure 6.2). On peut cloner des fragments de 5 à 10 Kb en utilisant les plasmides, 30 Kb avec les cosmides, et jusqu'à 1 Mb avec des *YAC* (chromosomes artificiels de levure). Toutefois, l'utilisation de ce vecteur peut générer des erreurs à cause d'échanges d'ADN. C'est pourquoi on préfère utiliser des *BAC* (chromosomes artificiels de bactéries) [21], qui permettent de cloner des portions d'une moyenne de 150 Kb. Ceci a permis de construire des banques, afin d'identifier les clones par méthode de *criblage* (*screening*). Il a fallu entre 5 et 7 ans pour l'identification du gène responsable de la mucoviscidose, et 15 ans pour celui de la chorée de Huntington.

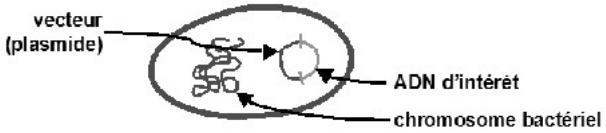


FIG. 6.2 – Le clonage grâce aux plasmides

Une fois que l'emplacement du gène a été déterminé, on cherche à en obtenir sa structure de nucléotides. On effectue donc un *séquençage*, par la méthode de Sanger [14]. L'ADN polymérase synthétise un brin à partir de la portion d'ADN à séquencer, mais dans un milieu contenant pas seulement les désoxyribonucléotides (dNTP : désoxyNucléotide TriPhosphate) habituels, mais aussi des didésoxyribonucléotides (ddNTP), par exemple, ddCTP, qui a été rendu fluorescent. La différence entre ddCTP et le nucléotide C est l'absence d'un groupement OH, qui fait que la synthèse du brin ne peut continuer après ce ddCTP. Ainsi, on va obtenir des brins se terminant tous par ddCTP, de longueur variable. On fait de même avec les trois autres ddNTP (fluorescents aussi mais identifiables les uns par rapport aux autres), puis on fait migrer tous ces brins, qui se positionnent donc les uns par rapport aux autres en fonction de leur longueur. Cette suite de bandes est ensuite analysée pour obtenir un *chromatogramme*, suite de pics de quatre couleurs différentes. Son analyse permet de déterminer la séquence des nucléotides. l'ABI PRISM 310 est une machine

qui réalise ce processus, qui permet donc de séquencer des brins d'un millier de bases.

C'est à la fin des années 80 que les méthodes de séquençage ont été appliquées à grande échelle afin d'obtenir une carte du génome humain. Sous l'impulsion de Daniel Cohen [22] et de Bernard Barataud [23] de l'Association Française contre les Myopathies, le Généthon a été créé en 1990. Jean Wessenbach y a créé une première carte génétique globale en utilisant 5000 marqueurs. Puis d'autres *genome centers* [44], comme le Génoscope d'Evry (Centre National de Séquençage, créé en 1997), ont été construits à la fin des années 90.

6.2.2 Les méthodes de reconstruction du génome

Il existe deux méthodes principales pour séquencer un génome entier [27] : shotgun, et clone-by-clone. A développer....

On peut aussi s'intéresser aux réarrangements du génome. A développer, images miroir, translocations....

Repérer les séquences répétées

Il faut en effet enlever les régions intergéniques : on peut donc éliminer les séquences répétées (les SINE et LINE), en utilisant l'algorithme de Smith Waterman, ou avec l'outil plus spécifique Repeatmasker, qui, en fait, marque les séquences en remplaçant les nucléotides par des `N`.

Le clustering et la création de contigs

Un contig est un assemblage de fragments successifs de morceaux d'un même chromosome qui se recouvrent partiellement. On peut utiliser les logiciels PHRAP (qui élimine les erreurs de séquençage, par exemple une séquence `GGGGGGG` finale indiquerait qu'il n'y avait plus de ddATP, ddCTP et ddTTP, avant d'utiliser un algorithme de Smith Waterman), CAP3 [25] et ARACHNE [6] et ARCHNE2 [31].

Prédiction des gènes

Pour les procaryotes, on utilise les outils GENMARK et GLIMMER. Pour les eucaryotes, on associe GENSCAN et GENOSCAN.

On utilise aussi des *puces à ADN* (*biochips* ou simplement *chips* en anglais) pour détecter la présence d'un brin d'ADN par appariement avec son complémentaire fixé sur la puce. Ces brins dans les puces sont des oligonucléotides, ils ne contiennent que peu de nucléotides qui ont pu être assemblées une par une. Certaines plaques peuvent actuellement contenir 30 000 puces.

Chapitre 7

Les microARN

Les *microARN* (abrégés *miRNA* en anglais) sont des petits brins d'ARN d'une vingtaine de nucléotides intégrés dans la tige double brin de structures d'ARN en forme d'aiguille à cheveux (*hairpin* en anglais).

Alors que les gènes à l'origine des microARN font partie des introns, c'est à dire de portions non codantes de l'ADN, ils ont un rôle important puisqu'ils régulent la traduction. En effet, un ou plusieurs microARN peuvent empêcher la traduction de plusieurs protéines. Ainsi, les effets peuvent être énormes : Arabidopsis, le végétal "classique"[1] change complètement de forme de feuilles en cas de surexpression de certains microARN, et n'est pas viable s'ils ne sont pas assez exprimés.

L'action de régulation des microARN intervient dans le noyau, juste avant le passage dans le cytoplasme et la traduction (soit avant la dernière étape de la figure 6.1, en anglais : *post-transcriptional gene silencing*). Après sa transcription, le microARN est en effet associé à un complexe *RISC* (*RNA induced silencing complex*), qui utilise la séquence du microARN comme détecteur. Dès qu'il repère la séquence complémentaire, il se fixe à l'ARNm débusqué, et le découpe de façon périodique. Les portions d'ARN qui résultent de ce découpage sont soit éliminés, soit utilisés comme microARN.

Ainsi, l'apparition des microARN a certainement découlé du processus de défense contre les virus, la *RNAi* (*RNA interference*), découverte chez les végétaux en 1990, qui met en jeu un *dicer* qui coupe l'ARNm en portions de 21 à 23 nucléotides [5].

Les microARN n'ont pas été découverts dans tous les organismes, mais seulement chez les eucaryotes. En 2005, une centaine de familles ont été identifiées chez Arabidopsis, 300 chez les végétaux, et un peu plus chez les animaux.

Pour détecter des microARN, et notamment de nouvelles familles, il faut commencer par les caractériser, puisque chercher seulement de courtes séquences répétées ne suffit pas : on a vu qu'elles abondaient dans le génome. Une caractéristique très forte est que la séquence cible du microARN est extrêmement bien conservée. On peut retrouver pour certains des copies exactes dans différentes espèces.

...

[1] Cette plante à fleur de petite taille a été entièrement séquencée dès 2000, et ce pour de multiples raisons (http://fr.wikipedia.org/wiki/Arabidopsis_thaliana), notamment, elle possède très peu d'ADN non codant et a un cycle de reproduction court.

Deuxième partie

Phylogénie

En complément de cette partie, vous pourrez là aussi vous référer au cours d'Infobiogen [10][2]. Le livre de référence en la matière, en anglais, est *Inferring Phylogenies* de Joe Felsenstein, qui a aussi effectué un recensement souvent mis à jour des programmes disponibles en phylogénie[3].

[2] http://www.infobiogen.fr/doc/documents.php?cours=phylogenie, lien vérifié le 29/06/2005

[3] http://evolution.genetics.washington.edu/phylip/software.html, lien vérifié le 29/06/2005

Chapitre 8

Les arbres phylogénétiques

Les arbres phylogénétiques sont des arbres (au sens informatique du terme) qui montrent les relations de parentés entre les espèces (les feuilles de l'arbre) ou d'autres entités (des gènes, par exemple) supposées avoir un ancêtre commun (les noeuds de l'arbre). Mais ils ont plus qu'une utilité historique, avec plusieurs applications médicales : identifier les souches d'un virus, leur origine commune, ou encore, pour le VIH, la rapidité de ses mutations chez un malade pour déterminer si le traitement est efficace...

Comprendre les liens entre espèces intéresse les biologistes, sutout les zoologistes et botanistes. Les anatomistes et morphologistes ont trouvé des analogies structurelles et ont regroupé des espèces en taxons (un groupe se situant à un niveau quelconque de la classification hiérarchique des êtres vivants : ordre, famille, tribu, genre, espèce, sous-espèce, variété... On parlera donc parfois de taxons quand on évoquera les feuilles des arbres phylogénétiques.). Ceci a tout d'abord été rendu possible grâce à la découverte de fossiles et leurs analyses en *histologie* (partie de l'anatomie qui étudie la formation, l'évolution et la composition des tissus des êtres vivants). Puis, depuis la fin des années 60, mais surtout dans les années 80, des séquences de gènes ont été accumulées et classées, ce qui a permis de construire de nouveaux arbres fondés sur les données génétiques (avec parfois des surprises et des corrections d'arbres précédemment établis). Un concept fort en phylogénie est celui d'horloge moléculaire : en supposant que les mutations se font au hasard, on considère que les longueurs de branche des arbres phylogénétiques sont proportionnelles aux durées écoulées.

On peut optimiser trois critères pour construire ces arbres : la *distance* (1967), la *parcimonie* (parsimony, 1966), et la vraisemblance (likelihood, 1971). Trouver l'arbre le plus parcimonieux est très coûteux en temps (NP-complet [20]), mais donne le résultat optimal. C'est pourquoi on ne cherche pas en général à examiner tous les arbres de façon exhaustive. On préfère utiliser un algorithme glouton qui tente d'optimiser un de ces trois critères.

On ne fait qu'évoquer ci-dessous les principes généraux de méthodes liées à ces trois critères. Les descriptions complètes de ces méthodes pourront être trouvées dans les articles cités en bibliographie.

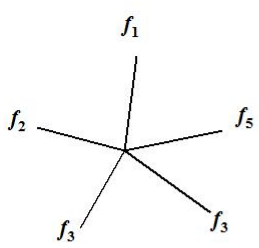
8.1 Optimisation de distance

8.1.1 UPGMA

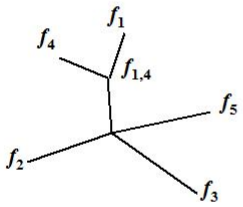
UPGMA (Unweigth Pair Group Method with Arithmetic mean) est l'algorithme utilisé pour construire le dendrogramme dans la seconde partie de l'algorithme de CLUSTAL en section 4.

On itère le processus suivant, illustré en figure 8.1, en partant d'un arbre en étoile :

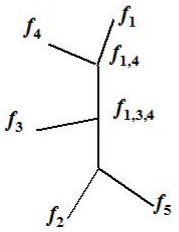
- On cherche dans la matrice le score le plus fort.
- On "joint" les deux séquences.
- On modifie la matrice de score, en supprimant la ligne et la colonne des deux séquences jointes, et en ajoutant la ligne et la colonne d'une séquence compromis des deux. Pour calculer cette ligne (colonne), on calcule la moyenne des deux lignes (colonnes) enlevées.



	f_1	f_2	f_3	f_4	f_5
f_1	0	5	3	1	12
f_2	5	0	7	5	5
f_3	3	7	0	5	8
f_4	1	5	5	0	12
f_5	12	5	8	12	0



	$f_{1\text{-}4}$	f_2	f_3	f_5
$f_{1\text{-}4}$	0	5	4	12
f_2	5	0	7	5
f_3	4	7	0	8
f_5	12	5	8	0



	f_2	$f_{3,1,4}$	f_5
f_2	0	6	5
$f_{3,1,4}$	6	0	10
f_5	5	10	0

FIG. 8.1 – L'algorithme UPGMA avec à chaque pas l'arbre phylogénétique et la matrice des distances correspondante.

Cet algorithme présente une limite : supposons qu'après une spéciation, un gène évolue beaucoup dans une des deux espèces, et pas énormément dans l'autre. Les deux séquences correspondantes ne seront pas très similaires, et celle provenant du gène ayant beaucoup évolué pourrait se retrouver non pas joint à son orthologue, mais vers la racine de l'arbre. UPGMA n'est donc fiable que dans le cas de séquences peu divergentes.

8.1.2 Neighbour-joining

L'algorithme de NJ (*neighbour-joining* [42]), aussi fondé sur la notion de distance, consiste à calculer les longueurs de branches de l'arbre et choisir l'arbre qui minimise la longueur totale. On part donc d'un arbre en étoile avec pour feuilles les séquences et pour noeud central un ancêtre commun supposé. On en calcule la longueur totale, puis on calcule pour la longueur totale de l'arbre pour la configuration de la figure 8.2 pour tout couple (i,j) de séquences. On choisit l'arbre $A_{i_{min},j_{min}}$ qui minimise cette longueur totale, et on réitère en considérant à chaque fois le le consensus du couple trouvé pour (i_{min}, j_{min}).

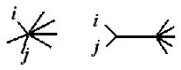


FIG. 8.2 – Arbre initial, puis après un pas de neighbour-joining

Ainsi, cet algorithme fondé sur la notion de distance permet de se débarasser du défaut d'UPGMA mentionné ci-dessus. Toutefois, la notion de distance elle-même n'est pas parfaite pour créer un arbre phylogénétique. Supposons en effet que l'on considère un brin d'ADN de 100 nucléotides où il y a eu 120 mutations. Si certaines mutations ont permis d'annuler la première mutation sur le même nucléotide, on aura l'impression qu'il y a eu moins de 100 mutations. Ainsi, la distance apparente entre deux codes génétiques n'est pas toujours la distance réelle en terme de mutations, surtout pour des espèces éloignées. C'est pourquoi il est nécessaire de réaliser des analyses de saturation.

8.1.3 Analyses de saturation

On va en effet essayer de contourner le problème en choisissant astucieusement les protéines à étudier en fonction des espèces dont on veut dresser l'arbre phylogénétique. Pour un arbre phylogénétique sur les humains, il faudra éviter les protéines avec grosses pressions de sélection et choisir celles qui sont divergentes. En revanche, pour un arbre contenant des mammifères et des mollusques, il faudra examiner des protéines très contraintes, comme celles qui se trouvent dans les ribosomes, afin d'obtenir un arbre significatif.

8.2 Optimisation de la vraisemblance

Cette méthode probabilistique consiste à calculer des vraisemblances d'arbres et choisir le plus vraisemblable. On peut procéder comme suit [17] : on part de deux espèces, et on ajoute les suivantes de la façon suivante : pour la k-ième espèce, on a $2k-5$ segments de l'arbre où elle pourrait s'accrocher. On calcule donc la vraisemblance de l'arbre pour chacune de ces configurations, puis on choisit la plus vraisemblable, avant d'ajouter l'espèce suivante.

Le calcul de la vraisemblance d'un arbre est détaillé en [17] et se fonde sur un calcul de vraisemblance arête par arête, qui lui-même utilise un calcul de vraisemblance nucléotide par nucléotide qui prend en paramètre le taux de substitution de nucléotides par rapport au temps.

8.3 Optimisation de la parcimonie

On peut par exemple tester tous les arbres possibles, et de trouver celui qui minimise le nombre de transformations élémentaires. Il faut toutefois réfléchir aux transformations élémentaires à considérer. En effet, il faut prendre en compte le code génétique. En effet, si une mutation de `Phe` vers `Glu` n'est qu'un changement d'acide aminé, le changement de codon correspondant fait passer de `UUU` à `CAA`, soit 3 mutations sur les nucléotides. De plus, la complexité étant exponentielle, cette méthode n'est à utiliser que pour classer un nombre très réduit d'espèces.

8.4 Enracinement d'un arbre phylogénétique

Les méthodes citées ci-dessus ne permettent pas nécessairement d'obtenir une phylogénie enracinée. On peut trouver la racine par la méthode d'*outgroup* ou celle du *midpoint* (point médian). La seconde consiste à repérer le noeud de l'arbre qui pourrait le mieux servir de consensus entre les séquences. La première est plus astucieuse : on ajoute à toutes nos séquences celle d'une espèce éloignée. Ainsi, elle se raccrochera à la racine de l'arbre phylogénétique que l'on avait obtenu. On réalise donc l'arbre phylogénétique avec toutes les séquences étudiées ainsi que cette séquence outgroup : la racine est le père de la séquence outgroup ajoutée.

8.5 Améliorations par réarrangements topologiques

Pour obtenir des algorithmes plus rapides, il est possible de partir d'un arbre construit rapidement, par exemple en utilisant le critère de distance, et de l'améliorer selon un autre critère. Pour ce faire, on peut essayer de réorganiser quelques parties de l'arbre, avec deux réarrangements : les NNI (Nearest Neighbour Interchange) et les SPR (Subtree Pruning Regrafting) illustrés en figure 8.3.

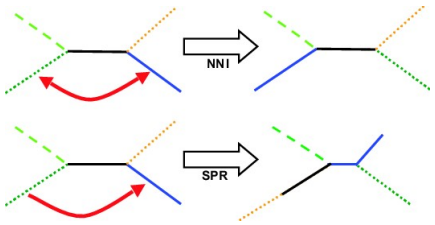


FIG. 8.3 – Réarrangements NNI et SPR

Un NNI consiste à échanger deux sous-arbres reliés par une arête, alors qu'un SPR consiste à détacher un sous-arbre, et l'agrafer sur une arête quelconque de l'arbre[1]. Ces mouvements sont particulièrement intéressants puisqu'ils permettent d'explorer l'espace des arbres phylogénétiques : il est possible de trouver une suite de mouvements NNI qui permettent de passer d'un arbre phylogénétique à tout autre ayant les mêmes feuilles. On peut donc aussi définir une

[1] On peut remarquer que l'ensemble des NNI est inclus dans celui des SPR.

distance entre deux arbres phylogénétiques ayant les mêmes feuilles [2, 8] : le nombre minimal de SPR ou de NNI qui a permis de passer de l'un à l'autre. La distance SPR est particulièrement intéressante puisqu'elle a une signification biologique : on peut l'utiliser pour modéliser les recombinaisons ou les transferts de gènes horizontaux.

8.6 Optimisation : du local au global

Comme on l'a vu à la section précédente, on peut, lorsque l'on essaie de maximiser la parcimonie sur un arbre phylogénétique ou plus généralement d'optimiser un critère sur un objet donné, modifier l'objet de façon locale et essayer, petit à petit, d'optimiser le critère. Un algorithme glouton consiste à choisir localement l'optimum du critère pour espérer optimiser l'optimum global. D'autres stratégies permettent de ne pas rester "coincés" dans des optimum locaux.

8.6.1 Le recuit simulé

Au lieu de choisir à chaque fois l'optimum local comme dans un algorithme glouton, on s'autorisera à chaque pas de l'algorithme, selon une certaine probabilité, de choisir une valeur moins bonne localement. Cette probabilité de choisir une valeur moins bonne décroit au cours du temps. Ainsi, on sera très flexibles au début, en espérant ne pas rester dans un maximmum local, mais plutôt s'approcher du maximum global, et là, dans les derniers pas, on finira de manière plus contraignante afin d'atteindre le maximum local, que l'on espère donc global.

Formellement, on a un paramètre de "température"[2] T qui décroit au cours du temps, et, en notant $V(i)$ la valeur à optimiser au temps i, si $V(i) > V(i-1)$, la valeur trouvée est meilleure, on garde les modifications effectuées au pas $i+1$. Si $V(i) \leq V(i-1)$, on garde les modifications effectuées au pas $i+1$ avec une probabilité $e^{\frac{V(i)-V(i-1)}{T}}$.

8.6.2 Le grand déluge

On ajoute un paramètre de niveau d'eau qui monte à chaque pas de l'algorithme, et oblige à rejeter les modifications si la valeur du critère est inférieure à ce niveau d'eau.

A préciser...

8.7 Les arbres de duplication

La moitié de l'ADN humain est constitué de séquences répétées. Parmi elles, on trouve des séquences répétées en tandem, c'est à dire des segments adjacents presque identiques, issus d'une suite de *duplications en tandem* d'un segment

[2] L'idée du recuit simulé est en effet inspirée du processus métallurgique de *recuit*, qui consiste à chauffer un métal, stabiliser sa température, puis le faire refroidir progressivement pour obtemir moins de dureté et plus de ductilité

original. Lorsque le motif répété est assez long (contient un gène), on considère que c'est la recombinaison inégale (figure 8.4) qui est à l'origine de cette duplication en tandem ([19], [41]).

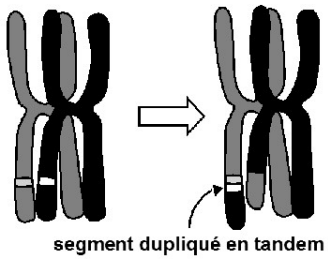


FIG. 8.4 – Recombinaison inégale pendant la prophase de la méiose

Ce processus est un mécanisme d'évolution des plus importants, puisque les gènes ainsi dupliqués, seront par la suite modifiés et pourront ainsi acquérir de nouvelles fonctionnalités. Reconstruire l'histoire des séquences dupliquées est donc fondamental pour comprendre leur évolution et leur fonction. L'outil de modélisation de cette évolution est l'*arbre de duplication*, qui permet de représenter la suite des événements de duplication subis pour obtenir la séquence finale, en étant éventuellement enraciné par le segment ancêtre. Ces arbres se distinguent en *arbres de duplication simple* si l'événement de duplication ne concerne toujours qu'un segment, et *arbres de duplication multiple* s'il existe des événements qui concernent plusieurs segments adjacents.

Définissons formellement les arbres de duplication enracinés (RDT, *rooted duplication trees*), de façon récursive. L'ensemble de ses feuilles est ordonné, par l'ordre des différentes feuilles sur le chromosome, donc un arbre de duplication sera un couple (T, O), T étant un arbre phylogénétique et O un ordre sur les feuilles. Pour toute paire (u, v) de O, nous utiliserons la notation $u \prec v$ pour exprimer que u est *adjacent* à v dans O. On définit une *cerise* (g, u, d) comme une paire de feuilles soeurs (séparées par un unique sommet u dans T) de T, (g, d), et nous appelons $C(T)$ l'ensemble des cerises de T. La définition récursive suit une procédure inverse à celle de l'évolution : elle cherche un *événement de duplication visible*, *agglomère* cet événement et vérifie si l'*arbre réduit* est un arbre de duplication.

(T, O) est un *arbre de duplication* (RDT) de racine ρ si et seulement si :

(i) (T,O) contient uniquement ρ, ou

(ii) il existe dans $C(T)$ une série de cerises (g_i, u_i, d_i), $(g_{i+1}, u_{i+1}, d_{i+1}), \ldots,$ (g_k, u_k, d_k) avec $k \geq i$ et $g_i \prec g_{i+1} \prec ... \prec g_k \prec d_i \prec d_{i+1} \prec ... \prec d_k$ dans O, telle que (T', O') soit un arbre de duplication de racine ρ, où T' est obtenu à partir de T en enlevant $g_i, g_{i+1}, ..., g_k, d_i, d_{i+1}, ..., d_k$, et O' est obtenu en remplaçant $(g_i, g_{i+1}, ..., g_k, d_i, d_{i+1}, ..., d_k)$ par $(u_i, u_{i+1}, ..., u_k)$ dans O.

Si $k > i$, on dit alors que l'on a *aggloméré* l'*événement de duplication multiple* $(u_i, ..., u_k)$, comme montré dans la figure 8.5. Si $k = i$, on dit que l'on a *aggloméré la cerise* (g_i, u_i, d_i), et le noeud u_i est un *événement de duplication simple*.

Cette définition récursive, dont une version similaire existe pour les arbres non enracinés permet de vérifier si une phylogénie quelconque est un arbre de

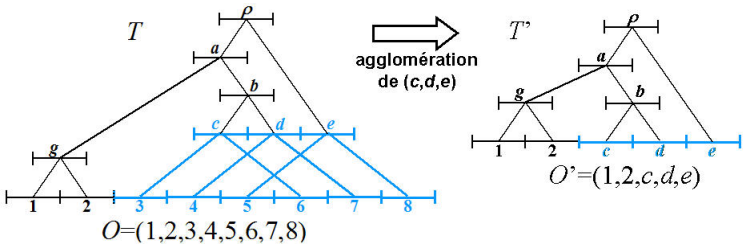


FIG. 8.5 – Agglomération d'un événement de duplication mutliple dans un arbre de duplication

duplication, en agglomérant chaque événement de duplication trouvé.

Si la reconstruction de l'histoire des duplications en tandem a été abordée par Fitch dès 1977 [19], les algorithmes pour l'automatiser sont assez récents. Retrouver l'arbre de duplication simple le plus parcimonieux est NP-complet [32], tout comme pour les phylogénies. On utilise donc des algorithmes heuristiques pour optimiser divers critères : la parcimonie [7] ou la distance [30].

Les arbres obtenus par ces algorithmes peuvent aussi être améliorés par l'application de méthodes de recherche locale [9]. Mais contrairement aux arbres phylogénétiques classiques, les réarrangements topologiques NNI et SPR (voir section 8.5) ne conservent pas la propriété d'abre de duplication. De plus, les NNI qui transforment bien un arbre de duplication en arbre de duplication, appelés *NNI restreints*, ne permettent pas d'explorer l'ensemble des arbres de duplication. En revanche, les *SPR* restreints le peuvent.

Chapitre 9

Le cas du VIH

Ce qui suit est un compte-rendu de la conférence de Simon Wain-Hobson donnée à l'ENS Cachan le 11 mai 2004 à propos du SIDA, axée plus particulièrement sur la variabilité du VIH et les conséquences sur son traitement. On y évoque notamment l'importance des recombinaisons dans la génétique de ce virus, alors que ce phénomène n'est pas pris en compte dans les algorithmes de reconstruction d'arbre phylogénétique, qui s'avèrent donc un modèle insuffisant.

9.1 Les origines du VIH

Parmi tous les virus existants, le VIH (virus de l'immunodéficience humaine) est le pire. En effet, le virus Ebola, fulgurant, s'éteint très vite, et la polio sera apparemment bientôt éradiquée.

Le virus est génétiquement très proche du VIS (Simian Immunodeficiency Virus) du singe. En Afrique Centrale, presque tous les singes sont porteurs du virus (24 espèces). On remarque dans l'arbre phylogénétique des VIH/VIS (figure 9.1) que le VIH1 a la même structure génétique que le VIS du chimpanzé, et que le VIH2 est très proche du VIS du mangabey enfumé (sooty mangabey).

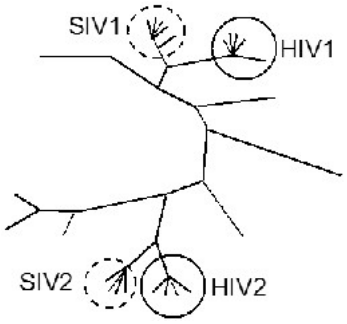


FIG. 9.1 – L'arbre phylogénétique du VIH

Les deux types de VIH ont chacun une origine géographique différente. Le VIH1 est originaire d'une zone aux vers l'Ouganda, le Burundi et le Rwanda,

et le VIH2 de l'Afrique Occidentale (figure 9.2). Il est étrange de constater que les deux virus se sont étendus à la même vitesse, et surtout au même moment : y aurait-il une raison pour que le virus se soit déclenché chez l'homme à un moment donné ?

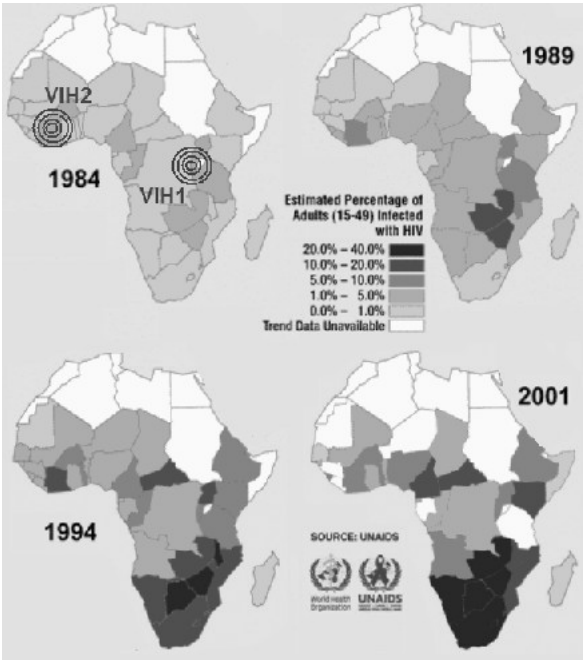


FIG. 9.2 – Le VIH en Afrique

Le virus VIS se transmet très bien à l'homme, et des tests récents ont montré que 15 à 20% des viandes de singe vendues dans les marchés des zones à risque étaient contaminées par le SIV... Il est donc admis que le VIS serait passé à l'homme par cette voie. Le virus a ensuite muté, et il existe actuellement plusieurs sous-types de VIH1. Le sous-type A représente 25%, le type B, 16%, présent surtout en Amérique du Sud, le C, apparemment le mieux transmis actuellement se trouve en Afrique du Sud, mais s'est étendu à l'Amérique du Sud et l'Asie. Le E se trouve en Asie du Sud-Est.

Une autre théorie a cependant été récemment proposée, prétendant que le SIDA aurait pu résulter de la vaccination contre la polio effectuée vers 1955 en Afrique Centrale, où des reins de chimpanzés auraient été utilisés pour réaliser ces vaccins. Le responsable de l'époque dément, et Monsieur Wain-Hobson a tenu à souligner que l'enquête du journaliste qui a construit cette théorie fourmillait d'erreurs et de citations tronquées, mais surtout qu'elle jetait le discrédit sur toute la communauté scientifique. Ces arguments sont développés dans un entretien qu'il a accordé à *l'Humanité* [34]. L'Institut Pasteur est donc actuellement en train de rechercher des traces éventuelles de VIH dans des organes datant d'avant 1955, ce qui prouverait l'invalidité de cette thèse. En effet, cette affaire doit être prise très au sérieux, puisqu'au vu de la catastrophe historique que sera le SIDA, il est nécessaire que ses origines soient clairement établies.

9.2 Les raisons de la difficulté de mise au point d'un vaccin

Tout d'abord, évoquons la structure du virus. C'est en fait un brin d'ARN, qui n'est copié en ADN qu'une fois dans la cellule, grâce à la *reverse-transcriptase*, c'est pourquoi on l'appelle *rétro*virus. Il contamine les lymphocytes (dès 12 heures après le premier contact avec le virus), qui sont alternativement actifs (en cas d'infection), puis mis au repos (en fin d'infection, figure 9.3). Le VIH s'adapte à ses phases de repos, devenant donc indétectable à ces moments-là. Voici la principale raison pour laquelle mettre au point un vaccin s'avère difficile.

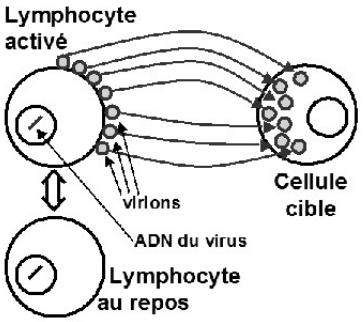


FIG. 9.3 – Le VIH et les lymphocytes

Habituellement, en effet, on utilise l'*immunité stérilisante*, c'est à dire que l'individu n'est pas malade alors que certaines de ses cellules sont tout de même infectées par le virus : l'infection est simplement contenue. Mais avec le VIH, impossible, puisqu'il lui suffit de se "cacher" dans quelques cellules, pour pouvoir se retransmettre en masse plus tard. Pour l'instant, les infections par rétrovirus sont donc considérées comme des infections à vie, et on pense qu'une vaccination servirait juste à réduire la transmission du virus.

De plus, la recherche sur les vaccins n'avance pas assez vite en pratique, principalement puisqu'on ne connaît pas encore les détails de l'action d'un vaccin. On ne peut donc que constater que les meilleurs résultats à ce jour sont ceux obtenus avec un virus vivant atténué. Mais cette solution est très risquée. Le virus pourrait en effet muter et devenir pathogène. De plus, le fait qu'il insère son ADN dans l'ADN chromosomique est aussi un problème. Mais les autres solutions testées semblent inefficaces : les anticorps neutralisants échouent, puisque les protéines virales sont "cachées" sous des glucides. Ainsi, l'affinité entre le VIH et le récepteur est plus de 100 fois plus grande que celle entre le VIH et l'anticorps. On essaie aussi de travailler sur le promoteur (paragraphe 6.1), puisque le changer permet d'atténuer significativement le virus, mais cette atténuation est telle qu'aucune réponse immunitaire ne se déclenche alors.

9.3 VIH et génétique

Les cellules infectées le sont toujours par plusieurs copies du virus. En effet, pour infecter l'ADN d'un lymphocyte, on estime que de 500 à 4000 virions sont nécessaires. Une cellule infectée présentre alors 29% d'acides aminés différents

par rapport à une cellule saine, ce qui équivaut à l'hétérogénéité entre les ADN de l'homme et du poulet, et montre au passage la robustesse du phénotype.

Mais le plus surprenant concernant le VIH est sûrement son caractère très recombinogène. Il n'est pas clônable, donc éphémère, et ceci empêche les généticiens de travailler dessus en détail. Il faut donc, tels les peintres impressionnistes, en saisir l'aspect général. Ceci n'est pas vraiment le cas actuellement. En effet, ce sont des méthodes qui ne tiennent pas compte des recombinaisons qui ont abouti à dater l'apparition du virus aux alentours de 1930. Il se pourrait en fait qu'il soit beaucoup plus jeune (1950 ?). Les logiciels classiques de construction d'arbres phylogénétiques ne considèrent pas non plus les recombinaisons (figure 9.4). Le programme SplitsTree permet d'en tenir compte, comme nous le verrons dans le chapitre 10.

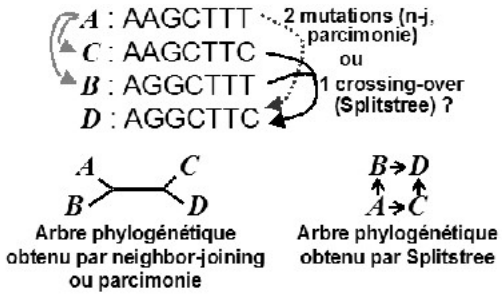


FIG. 9.4 – VIH et recombinaisons : arbre ou réseau phylogénétique ?

9.4 En conclusion

La variabilité du virus n'est donc pas un problème si dramatique, puisqu'il s'avère que l'on peut tout de même vacciner efficacement (avec des virus atténués). Le vrai problème en ce qui concerne la vaccination est celui de la non-élimination des lymphocytes contaminés au repos, qui implique une infection à vie. L'absence de traitement radical et définitif à court terme implique donc l'importance de la prévention contre le SIDA, qui pourrait sinon prendre des proportions inimaginables.

Chapitre 10

Les réseaux phylogénétiques

Le modèle d'arbre en phylogénie s'avère insuffisant pour représenter les transferts horizontaux de gènes, les recombinaisons ou les hybridations. Dans ce cas, on est amené à utiliser des réseaux phylogénétiques [**?**]. Ce terme générique regroupe plusieurs types de réseaux plus précis, en fonction justement du type de phénomène que l'on veut représenter parmi les trois cités ci-dessus, ou justement de l'indécision du phénomène mis en jeu : dans le cas où l'on sait juste qu'il y a un conflit entre différentes phylogénies, mais qu'on veut tout de même représenter les diverses évolutions possibles.

10.1 Le consensus d'arbres phylogénétiques

Pour obtenir des phylogénies fiables sur des espèces, il est préférable d'utiliser beaucoup de gènes, mais aussi le plus d'espèces possibles. Au lieu d'effectuer une approche de bas niveau en reconstituant l'arbre phylogénétique total en partant de toutes les séquences disponibles, on peut choisir une combinaison de haut niveau, qui consiste à combiner plusieurs arbres existants. C'est ce qu'on appelle le problème de superarbre (SuperTree).

Une introduction sur ce thème, avec présentation des algorithmes Build, MinCutSuperTree, RankedTree, ou la décomposition par quadruplets est disponible en [13].

Le programme Splitstree permet d'effectuer, grâce à la méthode Z-closure [29], un consensus sur des arbres n'ayant pas nécessairement les mêmes taxons, afin d'obtenir un réseau phylogénétique. En effet, on n'a pas ici de SuperTree, mais un SuperNetwork, puisque les conflits phylogénétiques font perdre la structure d'arbre. L'outil de base de cette technique de consensus est le *splits graph*, présenté à la section suivante.

10.2 La visualisation des conflits phylogénétiques par les splits graphs

L'outil utilisé ici, le splitsgraph, ressemble à un arbre phylogénétique par son aspect, mais est fondé sur le concept de *split*, c'est à dire de séparation de l'ensemble des taxons en deux groupes

10.3 La recherche des recombinaisons

...

10.4 La recherche des transferts horizontaux

Pour identifier des portions d'un génome provenant d'un transfert horizontal, il est possible d'utiliser les conflits phylogénétique. En effet, en choisissant astucieusement plusieurs familles de gènes dont on construit la phylogénie, si une des familles a été concernée par le transfert horizontal, on obtiendra un arbre différent pour cette famille. On distingue deux types de gènes : les primaires (parmi lesquels les *gènes domestiques* qui sont toujours exprimés dans toutes les cellules), qui sont essentiels, et les secondaires, qui moins nécessaires, et peuvent donc provenir de transferts horizontaux. Les 16S sRNA sont un exemple de séquences (d'ARN ribosomal) qui permettent de construire des bonnes phylogénies sur les espèces, et peuvent donc servir à construire l'arbre de référence que l'on va comparer avec l'arbre phylogénétique induit par les gènes secondaires.

Une seconde approche pourrait être de fouiller directement la séquence d'ADN à la recherche de spécificités des séquences dont on sait qu'elles sont issues d'un transfert horizontal.

Troisième partie

Modélisation, reconstruction et analyse de réseaux biologiques

Chapitre 11

Introduction aux réseaux biologiques

Ce domaine de la bioinformatique est assez récent. Il consiste à utiliser la masse de données diverses disponibles actuellement (des catalogues de gènes, l'état d'une cellule à un certain instant, certains mécanismes précis dans les cellules) pour reconstituer le fonctionnement général d'une cellule.

11.1 Le contexte de ce thème

Dans les années 80 se sont développées les technologies de biologie moléculaire. Les progrès informatique, autant du point de vue matériel que logiciel (algorithmes de comparaison prenant en compte les erreurs) ont permis la reconstitution du génome humain, entre 2001 et 2003. Nous avons donc à disposition une liste de gènes et de protéines, mais leur fonction n'est pas toujours élucidée. La structure tridimensionnelle des protéines a aussi été étudiées, mais les méthodes de prédiction se sont révélées moins efficaces que prévu. L'essor de la bioinformatique dans ces années 80 a donc parmis le passage de l'*hypothesis-driven research* (on part d'une hypothèse, on la valide en la testant) à la *data-driven research* (on part de données existantes, que l'on compare pour trouver des résultats)

Dans les années 90, on a travaillé sur les comparaisons de séquences, la phylogénie, l'assemblage de séquences, la prédiction de gènes ou de structures 3D de protéines afin d'effectuer des *annotations fonctionnelles*, c'est à dire élucider la fonction de chaque gène. Ces annotations fonctionnelles peuvent être obtenues de façon expérimentale sur des animaux, ou en étudiant les similarités de protéines, ce qui implique un degré de précision divers dans la connaissance des fonctions des protéines.

Ces deux décennies de recherche ont permis de lever des objections au dogme de la génétique qui consistait notamment à penser qu'un gène code pour une protéine qui a une fonction. En fait, certaines protéines agissent pour déterminer la production d'autres protéines, ce qui constitue un réseau. Les gènes sont donc collaboratifs, et on se demande si les 30000 gènes humains ne pourraient pas être regroupés en seulement 1000 circuits génétiques. On a observé le nombre

minimal de gènes pour un être vivant : environ 300, codant pour 12 fonctions cellulaires.

11.2 L'introduction des réseaux

Différents types de réseaux sont utilisés en bioinformatique :

- les réseaux de régulation (production de la protéine en fonction de sa concentration actuelle, déterminer quoi influence quoi), thème d'actualité.
- les réseaux métaboliques, biochimie classique, thème étudié dans les années 50.
- les réseaux d'interaction protéine/protéine
- les réseaux de signalisation ou signaling pathways (une molécule arrive, est reconnue par un récepteur, ce qui déclenche une réaction en chaîne).

Il faut en général choisir un type de modèle pour représenter le réseau à analyser, parmi les possibilités suivantes (pour des modèles discrets) :

- modèles orientés objets
- modèles fondés sur les graphes
- réseaux booléens
- noeuds de Petri (mais ça ne passe pas bien à l'échelle, *not scalable*)
- algèbre des processus (mais peut-être est-ce trop complexe)

Afin de préciser le modèle choisi, on est amené à déterminer ses caractères suivants :

- Est-il explicatif ?
- Est-il prédictif ?
- Est-il suffisamment complet, expressif ?
- Est-il assez simple ?
- Quels sont les outils théoriques associés ?

On peut alors choisir un outil de visualisation, comme BioCyc, KEGG, ou Cytoscope...

Le but final est donc de connaître parfaitement tous les mécanismes présentés dans le schéma 11.1.

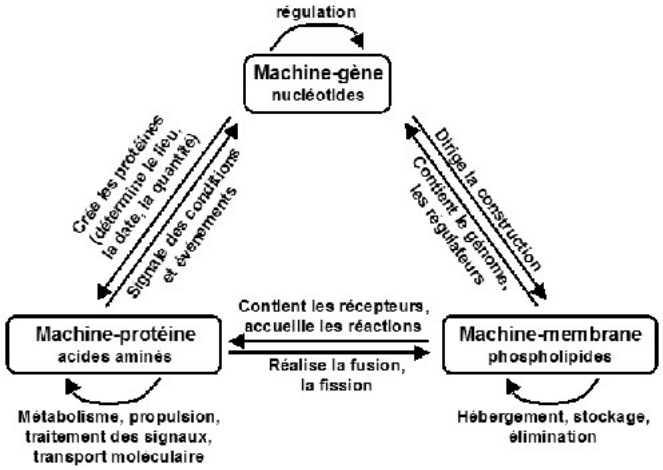


FIG. 11.1 – Modélisation d'une cellule en trois machines, d'après Luca Cardelli [11]

Chapitre 12

La cellule

12.1 Les caractéristiques de base de la vie

- Les atomes de la vie : C H O et N sont les quatre atomes qui constituent principalement les organismes vivants. Voici les proportions massiques pour les cellules animales :
  - carbone : 63%
  - hydrogène : 19%
  - oxygène : 9%
  - azote : 5%
- Le métabolisme :

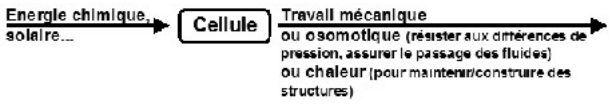


FIG. 12.1 – Le métabolisme d'une cellule

- Les fonctions du métabolisme :
  - le métabolisme énergétique : extraction et stockage de l'énergie (liaisons phosphate dans les molécules d'ATP)
  - le métabolisme intermédiaire : transformation des matières étrangères en de petites molécules.
  - les macromolécules : assemblage de petites molécules (*precursors*) en macromolécules
- La réaction aux stimuli externes, par exemple :
  - vision.
  - stimuli nerveux.
  - homéostase : se conserver selon les conditions de pression ou température.
- La reproduction

12.2 Les raisons de l'intérêt pour les cellules

- Tous les êtres vivants en ont (Schleiden et Schwann, 1938). La cellule est la structure fondamentale et fonctionnelle de la vie.
- Une chaîne de parenté relie toutes les cellules à une hypothétique cellule originelle : la vie est "continue" à travers les cellules.
- La biologie cellulaire, à l'origine, était descriptive, mais des techniques d'intervention sur la cellule, grâce à des concepts et méthodes importés de la biochimie ou de la biologie moléculaire ouvrent de nouveaux horizons.

Chapitre 13

Les bases de la reconstruction de réseaux

13.1 Les puces à ADN

Elles permettent l'étude instantanée de l'état des ARNm, dont la concentration est corrélée avec la concentration des protéines. Elles fonctionnent par hybridation de sondes avec la séquence ARNm cible, ou l'ADN obtenu par transcription inverse. Les sondes sont en fait des chaînes de quelques dizaines de paires de bases. Celles des puces Agilent (HP Inkjet Printing) contiennent 60 à 70 paires de bases, celles d'Affymetrix, 25 paires de bases. On ne peut donc pas avoir une sonde correspondant à un gène.

Pour fabriquer une puce, on traite successivement chaque couche pour toutes les sondes, en plaçant par exemple au début tous les `A` de la première couche pour toutes les sondes qui commencent par un `A`, puis tous les `C`, les `G`, les `T`, puis on passe à la seconde couche...

La séquence cible peut provenir de tissus différents, ou des mêmes tissus dans le même organisme (pour déterminer si la cellule cible est cancéreuse par exemple), ou des mêmes tissus dans des organismes différents (pour déterminer la fonction du gène cible). On peut aussi tester une même cellule cible à deux instants différents.

Pour réaliser les tests, on utilise soit une cible mobile sur les sondes fixes, soit le contraire. Après utilisation (Figure 13.1), la puce peut-être déshybridée et réutilisée 300 fois.

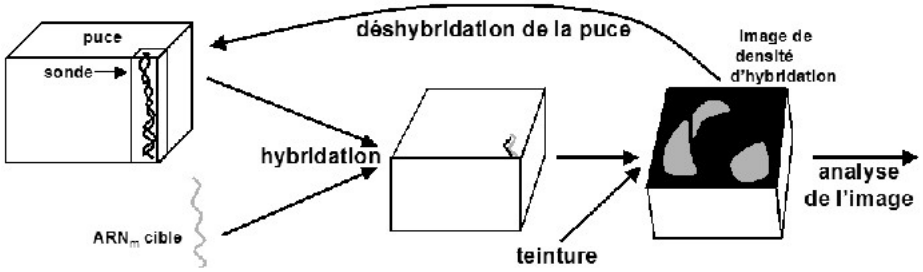


FIG. 13.1 – Les puces à ADN

L'image obtenue doit alors être analysée pour résoudre les problèmes suivants :

- *normalisation* : des caractéristiques de l'image (luminosité, grille) afin qu'on puisse comparer des images issues d'expériences ou de puces différentes (paragraphe 13.2.2).
- *classification* : partition de l'ensemble des conditions pour classifier un tissu : par exemple, est-il cancéreux ?
- *feature selection* : trouver un sous-ensemble de gènes caractéristiques pour cette classification.
- *clustering* : identifier un sous-ensemble de gènes se comportant de manière similaire (section 13.3).
- *annotation fonctionnelle* : annoter les gènes ou clusters.

13.2 Les analyses de bas niveau

Ce sont les analyses les plus éloignées des processus biologiques. Elles permettent d'obtenir à partir des données brutes (l'image) une *matrice de niveau d'expression*, avec les gènes sur les lignes, et les conditions sur les colonnes. On rencontre trois types de problèmes :

- *analyse d'image* : il faut réussir à extraire des résultats sans erreur.
- *agrégation de signaux de différentes sondes* : si plusieurs sondes sont utilisées pour détecter un seul gène, dans les puces à oligos par exemple.
- *normalisation* : évoqué ci-dessus.

13.2.1 Analyse d'image

Il s'agit tout d'abord de localiser la présence des sondes sur l'image obtenue. En effet, on n'a pas une grille claire et régulière, à cause de problèmes de scan par exemple. La méthode consiste à segmenter l'image et aligner localement avec des portions de grille. Une astuce consiste aussi à place sur les côtés de la grille successivement et périodiquement une sonde qui sera teintée, et une qui ne le sera pas.

Il faut aussi choisir les pixels les plus représentatifs, situés dans le milieu de la grille en général. Pour extraire l'intensité, il faut aussi choisir une méthode : prendre une moyenne ? Laquelle ? Il faudra de plus faire éventuellement quelques corrections locales d'intensité.

13.2.2 Normalisation

Il faut arriver à séparer les sources de variations de la luminosité entre variations biologiques et artefacts expérimentaux (préparation des échantillons, couleur et luminosité des teintures, paramètres du scan), et assurer qu'on puisse comparer entre elles des mesures issues de différents puces, ou dans des conditions différentes.

On doit choisir des gènes pour normaliser. On pourrait prendre tous les gènes présents sur chaque puce, en espérant qu'ils se comportent pareil en moyenne

dans les deux expériences, mais on normalisera de même des gènes peu ou beaucoup exprimés. Affymetrix a choisi 100 gènes souvent exprimés pour le calibrage, mais cette méthode est peu intéressante si l'on étudie des gènes qui apparaissent peu. On peut donc plutôt choisir de normaliser les distributions d'intensité.

Une autre alternative est de sélectionner les gènes après une première analyse : ceux qui s'expriment de la même manière dans les deux expériences sont utilisés pour la normalisation.

13.3 Clustering

On cherche regrouper des gènes se comportant de la même manière, en optimisant les deux critères suivants : homogénéité (pour les gènes regroupés) et séparation (entre les différents regroupements).

Pour ce faire [38], on peut réaliser un dendrogramme (section 8.1) afin de grouper les gènes en optimisant les distances.

On peut aussi utiliser la méthode des *K-moyennes* : on choisit tout d'abord le nombre de clusters voulu, K, puis on optimise le coût de tout clustering (somme des distances des gènes par rapport au centre de leur cluster). Cette méthode rappelle celles qui optimisent la vraisemblance pour la reconstruction de phylogénies (section 8.2).

Il est aussi possible d'utiliser des méthodes de cartes auto-organisatrices : on choisit une grille avec des noeuds représentants chacun un cluster, et on la "plonge" dans l'image résultat, en s'arrangeant pour que les noeuds de la grille correspondent bien à une moyenne des gènes de chaque cluster.

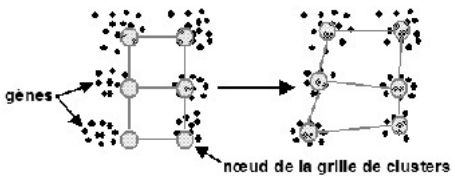


FIG. 13.2 – Construction d'une carte auto-organisatrice

Pour évaluer la qualité du clustering obtenu S, on peut calculer son homogénéité et sa séparation. On peut aussi calculer la distance (de Minkowski, D_M, ou de Jacquard, D_J) par rapport à un clustering de référence T. En notant n_{01}, le nombre de paires appartenant à T et pas à S, n_{00}, le nombre de paires appartenant à T et à S..., $D_M(T,S) = \sqrt{\frac{n_{01}+n_{10}}{n_{11}+n_{00}}}$ et $D_J(T,S) = \sqrt{\frac{n_{11}}{n_{11}+n_{10}+n_{01}}}$

Mais ces différents algorithmes ont un point commun : le nombre de clusters doit être fixé au départ. On peut utiliser des algorithmes plus récents qui éventuellement ne le nécessiteront pas.

Classification parallèle à noyaux (parallel classification with cores)

On va modifier un graphe de similitude (avec un score de similarité sur les arêtes), pour obtenir un graphe constitué de k cliques. L'algorithme pourra être testé en perturbant un graphe connu et en vérifiant si l'algorithme arrive à le retrouver.

13.4 Le biclustering

...

13.5 La reconstruction de réseaux de transcription

...

Chapitre 14

Calcul de membrane

Le *calcul de membrane* (Projective Membrane Calculus) est un exemple de l'utilisation de règles d'inférence pour modéliser un processus biologique.

14.1 Notations

Le système représenté en figure 14.1 peut être noté de la façon suivante : $(|a|), (|b|), (|c|), e$. Si l'on imagine que l'on plaque le système sur une sphère, on peut le considérer plus depuis e mais depuis a pour obtenir $a, (|(|b|), (|c|), e|)$. C'est ce que l'on appelle l'*invariant de bitonalité*. On peut donc représenter formellement ces systèmes par des graphes 2-coloriables (bitonaux) (figure 14.1). Rappelons qu'un graphe est 2-coloriable si et seulement si ses cycles sont tous de longueur paire.

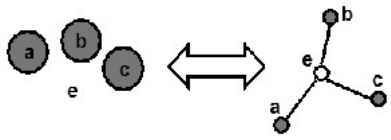


FIG. 14.1 – Représentations informelles et formelle de membranes

Les règles que nous fixerons pour agir sur ces systèmes devront respecter le bicoloriage de l'arbre. La règle de la figure 14.2 devra donc être interdite.

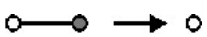


FIG. 14.2 – Règle invalide pour le bicoloriage de l'arbre

Enfin, contrairement à Luca Cardelli [11], nous considérerons des actions (fusions, fissions) orientées.

14.2 Syntaxe

14.2.1 Actions

$\sigma :=$ f_n ; fusion
$w_n(\sigma;\sigma)$; wrap
$b_n(\sigma;\sigma)$; bulle (exon int)

14.2.2 Branes

$\alpha :=$ 0
σ, σ ; produit
$\alpha.<\sigma,\sigma>$

14.2.3 Processus

$p :=$ $\Diamond$
$<\sigma;\sigma>(|P|)$; $<$*extérieur* ;*intérieur*$>$
P, P

14.2.4 Règles

Fusions :
$\frac{<b_n(\sigma';\tau'),\sigma;\tau>(|P|)}{<\sigma',\tau'>(|\Diamond|),<\sigma,\tau>(|P|)}$ drip, goutter
$\frac{<\sigma;\tau,b_n(\sigma';\tau')>(|P|)}{<\sigma,\tau>(|P,<\sigma',\tau'>(|\Diamond|)|)}$ pino (pinocytose)
Bulles :
$\frac{<f_n,\sigma;\tau>(|P|),<f_n,\sigma';\tau'>(|Q|)}{<\sigma,\sigma';\tau,\tau'>(|P,Q|)}$ mate
$\frac{<\sigma;f_n,\tau>(|<f_n,\sigma';\tau'>(|P|),Q|)}{<\sigma,\tau';\sigma',\tau>(|P,Q|)}$ exo (exocytose)
Wraps :
$\frac{<w_n,\sigma;\tau>(|P|)^B,<w_n^{\perp}(\sigma'';\tau''),\sigma';\tau'>(|Q|)^{A_1+A_2}}{<\sigma';\tau'>(|<\sigma'';\tau''>(|<\sigma;\tau>(|P|)_B|)_{A_1},Q|)_{A_2}}$ phago (phagocytose)
$\frac{<\sigma';\tau',w_n(\sigma'';\tau'')>(|<w_n,\sigma;\tau>(|P|),Q|)}{<\sigma'';\tau''>(|<\sigma;\tau>(|P|)|),<\sigma';\tau'>(|Q|)}$ bud

14.2.5 Versions informelle et bitonale des règles

Voir figure 14.3.

14.3 Equivalence projective

$<\sigma;\tau>(|P|), Q \sim P, <\tau;\sigma>(|Q|)$ (figure 14.4)

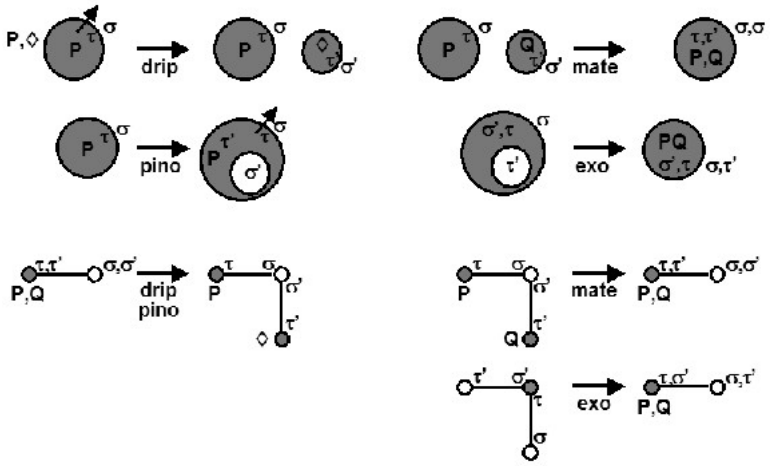


FIG. 14.3 – Règles en versions informelle et bitonale

FIG. 14.4 – Illustration de l'équivalence projective

Quatrième partie

Annexe

Chapitre 15

Mise en pratique sous Linux

15.1 Commandes de base

15.1.1 Commandes de gestion des dossiers

`passwd` : changer le mot de passe
`ls` : afficher dossier (`-l`, `-a`, `-s` : options, fichiers cachés, taille)
`cd` : changer dossier (`cd ..` : dossier parent)
`mkdir` : créer répertoire
`rmdir` : supprimer répertoire (attention : pas de poubelle)
`cp` *`fichier1 fichier2`* : copier
`mv` *`fichier1 fichier2`* (move) : déplacer
`rm` (supprime, `-fR` : options "forcer la suppression", "de façon récurrente")
`date` : date et heure
`df -k` : espace disque

15.1.2 Commandes de gestion des processus

`jobs` : tâches en cours
`top` : comme le CTRL ALT SUPP de XP
`ps -fu` *`Utilisateur`* : processus lancés par l'*`Utilisateur`* (ps : process)
`kill -9` *`numéroprocessus`* : tue le processus (*`numéroprocessus`* : PID, `-9` : violent, `-STOP` : arrêt proc, `-CONT` : reprendre proc)
ajouter `&` en fin de ligne pour lancer le processus en arrière-plan
CTRL z : arrête le processus et rend la main
CTRL c : kill !, bg : bascule le processus en arrière plan, fg pour le ramener au premier plan
appuyer sur q pour quitter un processus.
Les fenêtres Linux. Alt Control puis :
F1-F6 : 6 fenêtres ligne de commande
F7 : fenêtre graphique

15.1.3 Types de transferts de fichiers

simples transferts (ftp : non protégé / scp : protégé)
utilisation en tant qu'utilisateur référencé (telnet, attention : non crypté ! / ssh : crypté `ssh -l gambette ssh.dptmaths.ens-cachan.fr`)
affichage html (browser)
utilisation disques (nsf)
Xterm : calculs sur l'ordi distant, affichage sur l'ordi local

15.1.4 Transfert FTP shell

- `ftp ftp ://findel'adresseftp`
- login : anonymous
- passwd : adresse mail
- `binary` pour dire qu'on transfère du binaire et pas du texte
- pour envoyer des fichiers : `put` (plusieurs fichiers : multiput : `mput`)
- pour recevoir : `get` (`mget`) / Recevoir les fichiers dont le nom commence par *nt* : `mget nt*`.
- `?` donne toutes les commandes possibles dans le ftp
- pour parcourir les dossiers locaux en ftp : `lcd`
- pour afficher des commandes locales en ftp : `!` (Ex : `!ls -l nomdedossier`)

15.1.5 Compression

`gunzip`... (`gzip` : compresse les fichiers gz)
Fichiers tar : `tar xvf nomfichiertar.gz`
Fichiers Z : `compress` / `uncompress`
Fichiers bz, bz2 : `bzip2`, `bunzip2`

15.1.6 Permissions

`chmod a+rxw *` pour changer les modes
Est-ce un dossier / read-write-execute utilisateur / rwe groupe / rwe tous

15.1.7 Commandes de gestion des fichiers texte

Affichage d'un fichier : `cat nomdefichier`
`|` : "pipe" composer deux commandes
`cat nom de fichier | more` : aller vers le bas
`cat nom de fichier | less` : aller vers le haut ou vers le bas
`tr "a" "A"` : rechercher remplacer
Recherche et extraction d'une chaîne de caractères :
`grep -e "18" nomdefichier` : affiche la ligne du fichier qui contient 18 si elle existe.
Paramètres de `grep` :
`-A10` 10 lignes after
`-B20` 20 lignes before

`>Nom` : Enregistre le résultat dans le fichier `Nom`
`»Nom` : Enregistre le résultat dans le fichier `Nom`, en fin de fichier
`head -n50` : affiche le début d'un fichier (50 premières lignes)
`tail` : affiche la fin d'un fichier
`option -n` dans `cat` ou `grep` : affiche la numérotation des lignes.
`cut` : l'option `-d` définit les délimiteurs de champs, `-f` définit la case qui sera affichée.
Exemple : appliquée sur `"salut comment ca va"`, `cut -d" " -f2` affiche `"comment"`.
Attention, si l'on affiche les numéros de ligne, ils sont aussi pris en compte.
`expr $X` : affiche la variable X à l'écran

Boucle for
`X` : variable, `$X` : valeur de la variable
`wc -l` : nb de lignes, `-w` : nb de mots

Boucle while

```
a = 1 ;
while [$a -le $fin]
do
cat SEQ_""$a"".dat » $seq ;
a=$(($a+1)) ;
done
```

affectation : Y=$((18))
concaténation de chaînes : `""` ; Ex : `séquence""$i"".txt`

```
if ... ; then ... else ... fi
```

15.1.8 Outils de base

FASTACMD

Options :
- `-d` : banque de données
- `-s`, `-i` : adresse du fichier avec les "accession numbers"

BLAST

[1] Fichiers de sortie :
- liste des protéines avec : Accession number, puis détails ; expectation value : probabilité pour la protéine.
- liste des alignements, du premier au dernier acide aminé commun, avec des X pour le bruit de fond.

Voilà les options à ajouter à la commande : `nohup ./blastp -p` :
- `-d` : base de données
- `-i` : fichier d'entrée
- `-o` : fichier de sortie

[1] ftp://ftp.ncbi.nih.gov

- `-v 500` : nb résultats
- `-b 500` : nb résultats détaillés
- `-e 10` : "expectation value"

Les différents BLAST :
– PROT PROT : blastp
– ADN ADN : blastn
– ADN PROT : blastx
– PROT ADN : tblastn

NJPLOT

[2] Pour afficher les alignements multiples, dessiner des arbres phylogénétiques.

EMBOSS

[3] Open Reading Frames (phase ouverte de lecture) : `showORF` / `getORF`

HMMER

– `hmmbuild` pour construire le modèle de Markov depuis un alignement multiple.
– `hmmcalibrate` pour calibrer le modèle de Markov.
– `hmmsearch` pour rechercher dans la base des séquences reconnues.

15.2 Centres de recherche en bio-informatique

- Europe : EMBL (European Molecular Biology Laboratory, Heidelberg) : EMBOSS
- France : Génopole d'Evry (Infobiogen) : `http://www.infobiogen.fr` (base de données + logiciels), Institut Pasteur : `http://www.pasteur.fr`
- USA : NCBI (National Center for Bioinformatics) : `http://www.ncbi.nlm.org` / `ftp://ftp.ncbi.org` (Genbank)
- Japon : DBJ

[2] `http://pbil.univ-lyon1.fr/software/njplot.html`, lien vérifié le 29/06/2005
[3] `ftp://ftp.uk.embnet.org/pub/EMBOSS`

Chapitre 16

Lexique anglais/français

Un glossaire de termes du génie génétique avec leur traduction en anglais a été publiée au Journal Officiel [33]. Google peut aussi être utilisé comme glossaire, en tapant dans la barre de recherche *define* suivi du mot dont vous recherchez la définition. Ca fonctionne très bien pour les termes anglais et pas mal en français.

Ci-dessous se trouve un lexique anglais/français de termes rencontrés en bioinformatique

amino acid	acide aminé
deoxyribonucleic acid (DNA)	acide désoxyribonucléique (ADN)
ribonucleic acid (RNA)	acide ribonucléique (ARN)
enhancer	activateur
adenine	adénine
DNA polymerase	ADN polymérase
neighbour-joining	agglomération de voisins
allele	allèle
base pairing	appariement de bases
transfer RNA (tRNA)	ARN de transfert (ARNt)
messenger RNA (mRNA)	ARN messager (ARNm)
autosome	autosome [chromosome non sexuel]
gene library	banque de gènes
cDNA library	banque d'ADNc [ADN complémentaires]
hairpin loop	boucle en épingle à cheveux
reading frame	cadre de lecture [phase de lecture]
self-organizing maps	cartes auto-organisatrices
test cross	croisement test
cytosine	cytosine
frameshift	décalage du cadre de lecture
genetic fingerprinting	empreinte génétique [régions très variables du génome pour différencier les individus d'une même espèce]

splicing	épissage
standard deviation	écart type
hairpin	épingle à cheveux
fertilization	fécondation
housekeeping genes	les gènes domestiques
housekeeping genes	les gènes de ménage
a genus	un genre [dans la classification hiérarchique du vivant]
a greedy algorithm	un algorithme glouton
guanine	guanine
hybridization	hybridation
genetic information	information génétique
intergenic DNA	région intergénique
support vector machines [SVM]	machines à vecteurs de support
labelling	marquage
genetic marker	marqueur génétique [élément qui permet de différencier deux génotypes]
miRNA [micro-RNA]	microARN
mismatch	mésappariement
mitosis	mitose
midpoint	point médian
promoter	promoteur
purine	purine [pu]
pyrimidine	pyrimidine
breed	race
recombination	recombinaison [génétique]
simulated annealing	recuit simulé
wild type	type sauvage
sequencing	séquencage
TATA box	séquence TATA
strain	souche
telophase	télophase
termination	terminaison
translation	traduction
reverse transcriptase	transcriptase inverse
transcription	transcription
genetic transformation	transformation génétique
transgenic	transgénique
variety	variété
zygote	zygote

Bibliographie

[1] P. Allain : *Régulation de l'expression des gènes*, Les médicaments, http://www.pharmacorama.com/Rubriques/Output/Therapie_geniquea2.php, lien vérifié le 30/06/2005.

[2] B. L. Allen, M. Steel : *Subtree Transfer Operations and their Induced Metrics on Evolutionary Trees*, Annals of Combinatorics, 5 :1 13, 2001, http://www.math.canterbury.ac.nz/~mathmas/research/rearrangement.pdf, lien vérifié le 30/06/2005.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman : *Basic Local Alignment Search Tool*, Journal of Molecular Biology 215 (1990), 403-410, http://www-math.mit.edu/~lippert/18.417/papers/altschuletal1990.pdf, lien vérifié le 30/06/2005.

[4] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, D. Lipman : *Gapped BLAST and PSI-BLAST : a new generation of protein database search programs*, Nucleic Acids Research, vol 25 no 17 (1997), 3389-3402, http://www.bscbioinformatics.com/Fac/Lib/altschul1997.pdf, lien vérifié le 30/06/2005.

[5] F. Bacchus : *Implication de la machinerie de l'ARN interférence sur le silencing des gènes* (2004), http://www.eleves.ens.fr/home/bacchus/genetique.pdf, lien vérifié le 05/07/2005.

[6] S. Batzoglou, D. B. Jaffe, K. Stanley, J. Butler, S. Gnerre, E. Mauceli, B. Berger, J. P. Mesirov, E. S. Lander : *ARACHNE : A Whole-Genome Shotgun Assembler* (2002), http://www.genome.org/cgi/doi/10.1101/gr.208902, lien vérifié le 30/06/2005.

[7] G. Benson and L. Dong : *Reconstructing the duplication history of a tandem repeat*, in Proceedings of Intelligent Systems in Molecular Biology (ISMB 1999), 44-53. AAAI, http://tandem.bu.edu/papers/reconstructing.duplication.history.ismb.1999.pdf, lien vérifié le 30/06/2005.

[8] V. Berry : *Algorithmes de comparaison d'arbres étiquetés - application à la reconstruction de phylogénies*, Ecole Jeunes Chercheurs en Algorithmique et Calcul Formel, Montpellier, 4 au 8 Avril 2005, http://www.lirmm.fr/~ejc2005/EJC_PDF/beam-comparArb.pdf, lien vérifié le 30/06/2005.

[9] D. Bertrand, O. Gascuel : *Topological Rearrangements and Local Search Method for Tandem Duplication Trees*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 02, no. 1, pp. 15-28, January-March 2005, http://www.lirmm.fr/~w3ifa/MAAS/Articles-en-ligne/BG-TCBB-05.pdf, lien vérifié le 30/06/2005.

[10] Biogen : *Recherche dans les banques de données de séquences similaires à une séquence d'intérêt*, http://www.infobiogen.fr/doc/tutoriel/SIMIL/cours_tot.pdf, lien vérifié le 28/06/2005.

[11] L. Cardelli : *Languages and Notations for Systems Biology*, Conférence Mont St Michel (09/15/2004), http://research.microsoft.com/Users/luca/Slides/ExecutiveSummary20min-LanguagesforSysBio.ppt.

[12] T. Cormen, C. Leiserson, R. Rivest, C. Stein : *Introduction à l'Algorithmique*, Dunod (2002).

[13] A. Criscuolo : *Comment combiner les données génomiques, ou le superarbre, cas particulier d'une plus vaste problématique*, http://www.lirmm.fr/~criscuol/superarbre.htm, lien vérifié le 05/07/2005.

[14] M. Delarue et G. Furelaud : *Dossier : le séquençage d'un ADN*, http://www.snv.jussieu.fr/vie/dossiers/sequencage/sequence.htm, lien vérifié le 28/06/2005.

[15] S. R. Eddy : *Hidden Markov Models*, Current Opinion in Structural Biology 6 (1996), 361-365.

[16] J. Fajkus, E. Sýkorová, A. R. Leitch : *Techniques in plant telomere biology*, http://www.ibp.cz/labs/LDMC/BioTechF05.pdf, lien vérifié le 29/06/2005.

[17] J. Felsenstein : *Evolutionary Trees from DNA Sequences : A Maximum Likelihood Approach*, Journal of Molecular Evolution 17 (1981), 368-376.

[18] J. Felsenstein : *Inferring Phylogenies*, Sinauer (August 2003), 580 pages.

[19] W.M. Fitch : *Phylogenies constrained by crossover process as illustrated by human hemoglobins in a thirteen-cycle, eleven amino-acid repeat in human apolipoprotein A-I*, Genetics 86 (1977), 623-644, http://www.genetics.org/cgi/content/abstract/86/3/623, lien vérifié le 30/06/2005.

[20] L.R. Foulds and R.L. Graham : *The Steiner problem in phylogeny is NP-complete*, Advances in Applied Mathematics 3 : 43-49 (1982).

[21] G. Furelaud, Y. Esnault (Genoscope) : *Dossier : le séquençage des génomes*, http://www.snv.jussieu.fr/vie/dossiers/genomes/, lien vérifié le 28/06/2005.

[22] N. Givernaud, J.-F. Picard : *Entretien avec Daniel Cohen*, 21/01/2002, http://picardp1.ivry.cnrs.fr/CohenD.html, lien vérifié le 28/06/2005.

[23] N. Givernaud, J.-F. Picard : *Entretien avec Bernard Barataud*, 07/06/2001, http://picardp1.ivry.cnrs.fr/BarataudB.html, lien vérifié le 28/06/2005.

[24] D. G. Higgins, P. M. Sharp : *CLUSTAL : a package for performing multiple sequence alignment on a microcomputer*, Gene 73 (1988), 237-244.

[25] X. Huang, A. Madan : *CAP3 : A DNA Sequence Assembly Program*, (1999).

[26] R. Hughey, A. Krogh : *Hidden Markov models for sequence analysis : extension and analysis of the basic method*, Computer Applications in the Biosciences, vol 12 no 2 (1996), 95-107.

[27] D. H. Huson, K. Reinert, S. A. Kravitz, K. A. Remington, A. L. Delcher, I. M. Dew, M. Flanigan, A. L. Halpern, Z. Lai, C. M. Mobarry, G. G. Sutton and E. W. Myers : *Design of a compartmentalized shotgun assembler for the human genome*, Bioinformatics Vol. 17 no. 90001

(2001), pages S132-S139, http://bioinformatics.oxfordjournals.org/cgi/content/abstract/17/suppl_1/S132, lien vérifié le 28/06/2005.

[28] D. H. Huson : *Algorithms in Bioinformatics II, lecture at Tübingen University* (Summer 2005), http://www-ab.informatik.uni-tuebingen.de/teaching/ss05/abi2/welcome.html, lien vérifié le 28/06/2005.

[29] D. H. Huson, T. Dezulian, T. Klöpper and M. A. Steel : *Phylogenetic Super-Networks from Partial Trees*, IEEE/ACM Transactions in Computational Biology and Bioinformatics, 1(4) :151-158 (2004), http://www-ab.informatik.uni-tuebingen.de/publications/papers/zclosure.pdf, lien vérifié le 05/07/2005.

[30] O. Gascuel, D. Bertrand, and O. Elemento : *Reconstructing the duplication history of tandem repeated sequences*, in O. Gascuel, Mathematics of Evolution and Phylogeny. Oxford University Press (2005).

[31] D. B. Jaffe, J. Butler, S. Gnerre, E. Mauceli, K. Lindblad-Toh, J. P. Mesirov, M. C. Zody, E. S. Lander : *Whole-Genome Sequence Assembly for Mammalian Genomes : Arachne 2*, 2003, http://www.genome.org/cgi/doi/10.1101/gr.828403, lien vérifié le 28/06/2005.

[32] D. Jaitly, P. Kearney, G. Lin, and B. Ma : *Methods for reconstructing the history of tandem repeats and their application to the human genome*, in Journal of Computer and System Sciences (2002), 65, 494-507, http://www.bioinformatics.uwaterloo.ca/papers/02deep.ps.

[33] L. Jospin, H. Nallet : *Arrêté du 14 septembre 1990 relatif à la terminologie du génie génétique*, Éducation nationale, jeunesse et sports ; Agriculture et forêt - NOR : AGRD9001622A - JO du 26-09-1990, http://www.dsi.cnrs.fr/RMLR/textesintegraux/volume4/43-adu14-09-1990(2).htm, lien vérifié le 29/06/2005.

[34] L'Humanité : *Ce n'est pas quand une thèse devient désagréable qu'il faut la cacher, entretien avec Simon Wain-Hobson*, 23/04/2004, http://www.humanite.presse.fr/journal/2004-04-23/2004-04-23-392464, lien vérifié le 28/06/2005.

[35] V. Makarenkov, D. Kevorkov, P. Legendre : *Phylogenetic Network Construction Approaches*, http://www.labunix.uqam.ca/~makarenv/articles/MKL_article.pdf, lien vérifié le 28/06/2005.

[36] L. Nahkleh : *Phylogenetic Networls*, PhD thesis (2004).

[37] S. B. Needleman, C. D. Wunsch : *A general Method applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins*, Journal of Molecular Biology 48 (1970), 444-453.

[38] S. S. Parthasarathy : *Data Mining in Bioinformatics : a gentle introduction to clustering*, http://genome.osu.edu/ibgp730/lectures_2003/Clustering.ppt, lien vérifié le 28/06/2005.

[39] W. R. Pearson, D. J. Lipman : *Improved tools for biological sequence comparison*, Proceedings of the National Academy of Sciences, 85 (April 1988), 2444-2448.

[40] T. F. Smith, M. S. Waterman : *Identification of common molecular Subsequences*, Journal of Molecular Biology 147 (1981), 195-197.

[41] G. P. Smith : *Evolution of repeated DNA sequences by unequal crossover*, Science 191 (1976), 528-535.

[42] N. Saitou, M. Nei : *The Neighbor-joining Method : A New Method for Reconstructing Phylogenetic Trees*, Molecular Biology and Evolution 4(4) (1987), 406-425, http://ftp.cse.sc.edu/bioinformatics/papers/3243saitounei.pdf, lien vérifié le 30/06/2005.

[43] V. Zoonekynd : *Algorithmique et bioinformatique*, http://zoonek2.free.fr/UNIX/49_bioinfo/Cours.html, lien vérifié le 28/06/2005.

[44] *Liste des centres de séquençage du génome (site du NCBI)*, http://www.ncbi.nlm.nih.gov/genome/seq/HsCenters.html, lien vérifié le 28/06/2005.

Index

acides aminés, 16
activateur, 25
agglomération de voisins, 37
aiguille à cheveux, 29
alignement, 13
alignement de séquences, 13
alignement multiple, 21
analyse de séquences, 9
arabidopsis, 29
arbre de duplication, 39
arbre phylogénétique, 35

bases de données, 13
BLAST, 19, 69
BLOSUM, 17

calcul de membrane, 61
carte du génome, 28
cat, 68
chromatogramme, 27
clonage positionnel, 26
CLUSTAL, 21

dendrogramme, 21, 22
désoxyribonucléotides, 27
didésoxyribonucléotides, 27
distance entre arbres phylogénétiques, 39
dot-plot, 13
duplication, 22
duplications en tandem, 39

EMBOSS, 70
enhancer, 25
exon, 25

FASTA, 19, 69
Felsenstein, 33

gène, 25
gènes indépendants, 27
gènes liés, 27
gènes domestiques, 48
génétique inverse, 26
génome, 25
glossaire, 71
glouton, 39
graine, 20
grand déluge, 39
great deluge, 39
grep, 68
Généthon, 28

hairpin, 29
histologie, 35
HMMER, 24, 70
HSP, 20
hydrophobe, 16
hydrophyle, 16

recombinaison inégale, 40
intron, 25

lexique français anglais, 71
likelihood, 35
LINE, 25

matrice BLOSUM, 17
matrice de score, 13, 16
matrice de transition transversion, 16
matrice PAM, 17
maximum segment pair, 20
microARN, 29
modèles de Markov cachés, 24
morgan, 27
motifs, 23
MSP, 20

Neighbour-joining, 37
NJPLOT, 70

orthologue, 22

PAM, 17

paralogue, 22
partie codante, 25
phénotype, 26
phylogénie, 31
profils, 23
promoteur, 25
protéines TF, 25

recuit simulé, 39
région de régulation, 26
répresseur, 25
réseau phylogénétique, 47
RISC, 29
RNAi, 29
rétrovirus, 45

Sanger, 27
score, 13
séquençage, 27
sida, 7
silencer, 25
SINE, 25
split, 47
spéciation, 22

séquences répétées en tandem, 25
taxon, 35
télomères, 25
transcriptase inverse, 45
transcription factor, 25

UPGMA, 36

vraisemblance, 35

Wikipedia, 8