Erwan HAUVUY
Benjamin LEBRAVE
Pierre NEUVIAL

Analyse statistique du lien entre les plages homogènes de séquences d'ADN de différents organismes

Groupe de Travail

Suivi par Pierre-Yves BOURGUIGNON et Bernard PRUM

Table des matières

Introduction

Le séquençage du génome est une expression choyée par les médias et par la communauté scientifique dans son ensemble. C'est aussi un concept propice aux engouements populaires, sur le compte duquel se fondent de grands espoirs ; tout récemment par exemple, le séquençage du génome du SRAS a suscité le plus grand émoi, alors même qu'il s'agit d'une étape essentielle certes, mais encore bien éloignée de la découverte d'un traitement de la pneumonie atypique.

En fait, derrière le terme séquençage se cache une quantité de travail très longue et souvent fastidieuse, que de nombreux chercheurs espèrent pouvoir systématiser, notamment par des méthodes mathématiques et statistiques, afin de gagner en moyens humains et matériels, et donc en temps.

Les génomes ont une structure universelle qui se prête bien à l'analyse statistique ; ils comportent un grand nombre de bases, les unités de codage du vivant, allant de quelques millions pour les espèces les plus primaires (bactéries), à quelques milliards pour les organismes les plus complexes (le génome de l'homme contient environ 3 milliards de base).

Le séquençage du génome d'une espèce permet d'abord d'approfondir la connaissance scientifique d'un organisme. Or les espèces sont liées entre elles, les lois de l'évolution couplées à l'observation biologique permettent de les classifier et d'établir différents niveaux de parentés. Le biologiste peut donc choisir d'analyser les génomes non pas de façon uniquement individuelle, mais il peut encore essayer de faire apparaître des similitudes entre génomes.

Au vu de l'universalité des mécanismes génétiques, il semble donc envisageable que l'application de méthodes probabilistes, qui permettent d'établir une représentation statistique des données, puisse souligner les similitudes et les différences biologiques entre génomes.

En particulier, l'identification de certaines propriétés pourrait éventuellement autoriser l'emploi de méthodes statistiques de prédiction de la structure d'un génome au moins partiellement inconnu.

Chapitre 1

Présentation

1.1 Cadre

Le sujet et les méthodes de notre groupe de travail s'inscrivent dans le domaine de recherche du laboratoire " Statistique et Génome " (LSG) dirigé par le Pr Prum. Situé au sein du Génopole d'Evry, cette équipe comporte une quinzaine de chercheurs et thésards dont les travaux gravitent autour de la résolution mathématique de problèmes de nature biologique. Plus spécifiquement, le gros de leurs recherches se situe dans la modélisation de séquences génétiques par chaînes de Markov et chaînes de Markov cachées (CMC, ou Hidden Markov Models, HMM).

Avant d'aller plus loin, quelques rappels biologique peuvent être utiles.

1.1.1 Quelques rappels de génétique

L'information qui détermine les caractéristiques biologiques d'un être vivant est contenue dans son génome. Dans chaque cellule d'un organisme, on trouve un ensemble de chromosomes, composés de deux brins d'Acide DésoxyriboNucléique (**ADN**). A l'échelle moléculaire, ils apparaissent comme les montants d'une échelle, liés entre eux par des barreaux.

Ces barreaux sont faits de deux **bases** et de la liaison chimique qui les relie. Les bases sont des molécules qui constituent un alphabet dans lequel s'écrit une séquence génétique. Ces bases sont :

- l'adénine **a**
- la cytosine **c**
- la guanine **g**
- la thymine **t**

Les couples de bases se faisant face sur les deux brins d'ADN ne peuvent être que de deux types : **a** et **t**, ou **c** et **g**.

Sur un brin donné, les bases peuvent être groupées par segments, en particuliers par **gènes**. La définition que nous avons retenue dans ce mémoire est la suivante : un gène est une séquence d'ADN qui contient toute l'information nécessaire à la synthèse d'une protéine. Cette définition sous-entend qu'un gène peut contenir des segments qui n'apportent pas directement d'information pour coder la protéine ; nous y reviendrons plus en détail.

Un gène est d'abord traduit en Acide RiboNucléique (ARN), molécule intermédiaire qui est ensuite transcrite en acides aminés pour former une protéine.

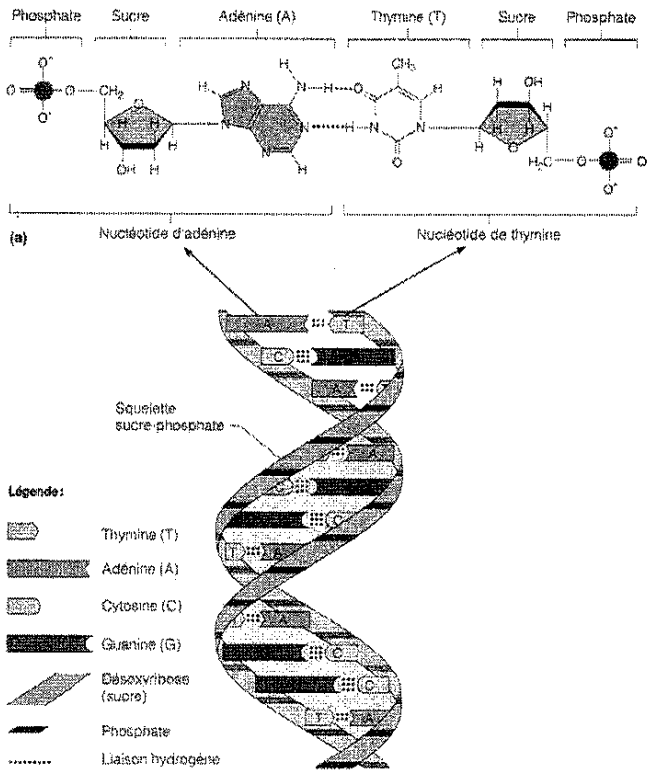

Figure 1.1: Structure de la molécule d'ADN.

Les bactéries contiennent un unique chromosome circulaire, par opposition aux chromosomes doubles en forme de croix tels qu'ils existent chez les organismes eucaryotes [1]. Là où le début du code d'un chromosome semble facile à déterminer chez ces derniers, il n'existe au contraire pas de véritable début ou de fin du génome d'une espèce de bactérie.

Chaque espèce est définie par un ensemble de caractéristiques biologiques, qui proviennent elles-mêmes des spécificités de son code génétique. Elle possède donc des gènes qui la définissent biologiquement. Ces gènes peuvent lui être propres ou être partagés avec un ensemble d'autres espèces.

Hormis les gènes à proprement parler, il existe divers types de segments dans une séquence d'ADN ; en particulier, on retiendra quatre distinctions.

Certaines parties d'une séquence ne sont pas utilisées directement pour coder une protéine, on parle alors de segments **non-codants**, par opposition aux segments **codants**. Chez l'homme, 90% du code génétique est non-codant, alors que pour une bactérie, 90% du code est codant. L'ensemble des gènes est donc confondu avec le codant.

[1] les eucaryotes sont les organismes dont les cellules ont un noyau, par opposition aux procaryotes, organismes primaires dont le noyau n'est pas délimité. Les animaux, les végétaux et les champignons sont des eucaryotes, tandis que les bactéries ou les virus sont des procaryotes.

Puisqu'un chromosome est constitué de deux brins, et du fait de la complémentarité des bases, on peut déduire parfaitement une séquence à partir de la séquence correspondante sur l'autre brin. En particulier, à chaque gène relevé sur un brin correspond sur le brin complémentaire une copie "en négatif" (les a sont transformés en t, les c en g, etc). Au sein d'une région codante, on parlera donc de segments **codants positifs** et **codants négatifs**.

Lorsqu'on considère un gène, on peut établir une nouvelle distinction suivant sa fonction : un gène peut coder une protéine qui assure soit une fonction essentielle de la vie (transport de l'oxygène, traduction du code génétique, certaines réactions immunitaires), soit une fonction spécifique à l'espèce ou à une famille d'espèces (le sexe chez les espèces sexuées, la photosynthèse chez les plantes, la couleur des yeux chez l'homme). On parle alors de gène dit **universel** ou **ubiquitaire** et de gène **spécifique**.

Notons que cette classification n'est en réalité pas duale, puisqu'hormis les fonctions réellement universelles, la fonction d'une protéine peut être propre à une espèce, ou à un ensemble d'espèces, voire à une grande famille du vivant. D'ailleurs, le choix des exemples ci-dessus tente de refléter ces différents niveaux de spécificité.

Quelle que soit la fonction biologique d'un gène, on peut différencier en son sein un ensemble de segments dits **exons** et **introns** : chez les eucaryotes, un gène peut être découpé en un ensemble de segments dont certains ne codent jamais une protéine, les **introns**, tandis que d'autres codent une ou plusieurs protéines, les **exons**. En d'autres termes, un gène peut être découpé en introns, qui n'interviennent jamais directement dans la synthèse de protéine, et en exons, dont les différents assemblages possibles entraînent la fabrication d'un ensemble de protéines différentes.

Par exemple, considérons la séquence suivante :

$$\underline{exon1}\,\overline{intron1}\,\underline{exon2}\,\overline{intron2}\,\underline{exon3}$$

Dans un premier temps, les introns peuvent être éliminés :

$$\underline{exon1\,exon2\,exon3}$$

En plus de la protéine associée à la séquence ci-dessus, les assemblages partiels de ces exons aboutissent à six séquences codantes différentes, donc à six autres protéines possibles :

$\underline{exon1}$ $\underline{exon2}$ $\underline{exon3}$ $\underline{exon1exon2}$ $\underline{exon2exon3}$ $\underline{exon1exon3}$

Bien sûr, un tel type de découpage est possible sur une séquence comportant un grand nombre d'introns et d'exons. Ceci explique comment le code génétique humain, qui ne comprend qu'environ 30 000 gènes, puisse aboutir à la synthèse de près de 150 000 protéines différentes.

Remarquons toutefois que cette distinction ne sera pas exploitée dans la suite de ce mémoire, étant donné que ces mécanismes d'assemblages n'existent pas chez les bactéries. Au contraire, la séquence codante correspondant à un gène ne peut être découpée pour obtenir d'autres gènes.

A l'aide d'un ou plusieurs de ces critères, on peut découper une chaîne donnée en différentes régions que nous appellerons **régimes biologiques**. En ce qui nous

concerne, les données dont nous disposons nous contraignent à nous restreindre à trois régimes : le non codant, le codant positif et le codant négatif. Mais des distinctions plus fines sont envisageables et mériteraient d'être étudiées. Nous précisons que les régimes sont biologiques car nous voulons les distinguer des **régimes statistiques** fournis par les chaînes de Markov cachées. Cette notion sera définie rigoureusement par la suite. Pour l'instant, disons simplement qu'il s'agit de régions de la chaîne qui sont choisies de manière à présenter une forte homogénéité statistique interne.

1.1.2 Séquençage et annotation d'un génome

L'analyse d'un génome comporte deux étapes principales. Dans un premier temps, il faut traduire le code génétique en une séquence informatique ; c'est le **séquençage**. Pour ce faire, la démarche est la suivante : le biologiste utilise plusieurs séquences identiques, qu'il découpe en segments élémentaires comprenant environs 500 bases, de sorte que les segments issus des différentes séquences se chevauchent. Les *séquenceurs* sont des appareils capables alors de déterminer la suite de bases composant ces segments. Ensuite, il faut les recoller entre eux jusqu'à ce qu'on obtienne la séquence initiale en un seul morceau.

Dans un deuxième temps, le traitement de la séquence est purement informatique : la séquence est lue par l'outil informatique dans le but d'identifier les gènes. C'est l'**annotation**. Pour cela, il faut analyser les motifs de la séquence ; la plupart des triplets de bases assure la synthèse d'un acide aminé[2].

aaa : Phénylalanine	gaa : Leucine	taa : Isoleucine	caa : Valine
aag : Phénylalanine	gag : Leucine	tag : Isoleucine	cag : Valine
aat : Leucine	gat : Leucine	tat : Isoleucine	cat : Valine
aac : Leucine	gac : Leucine	tac : Méthionine	cac : Valine
aga : Sérine	gga : Proline	tga : Thréonine	cga : Alanine
agg : Sérine	ggg : Proline	tgg : Thréonine	cgg : Alanine
agt : Sérine	ggt : Proline	tgt : Thréonine	cgt : Alanine
agc : Sérine	ggc : Proline	tgc : Thréonine	cgc : Alanine
ata : Tyrosine	gta : Histidine	tta : Asparagine	cta : Asparagine
atg : Tyrosine	**gtg** : Histidine	ttg : Asparagine	ctg : Asparagine
att : STOP	gtt : Glutamine	ttt : Lysine	ctt : A. Glutaminique
atc : STOP	gtc : Glutamine	ttc : Lysine	ctc : A. Glutaminique
aca : Cystéine	gca : Arginine	tca : Sérine	cca : Glycine
acg : Cystéine	gcg : Arginine	tcg : Sérine	ccg : Glycine
act : STOP	gct : Arginine	tct : Arginine	cct : Glycine
acc : Tryptophane	gcc : Arginine	tcc : Arginine	ccc : Glycine

Tableau 1: Code génétique.

Ce tableau représente le code complémentaire, c'est-à-dire le code correspondant à la séquence transcrite en ARN. On remarque dans ce tableau que certains triplets sont appelés **codons stop** (att, atc et act). Lorsqu'un codon stop est identifié,

[2]les acides aminés sont les briques à partir desquelles est assemblée une protéine.

on remonte la séquence jusqu'à ce qu'on trouve le premier codon du gène associé. Cette identification est plus difficile, car le codon **initiateur** tac code également la méthionine. Par conséquent, lorsqu'on cherche le premier codon d'un gène, on trouve le plus souvent plusieurs candidats. Généralement, on se fixe donc une longueur de gène minimale, de l'ordre de 200 bases, à partir de laquelle on recherche un codon initial.

Enfin, lorsqu'on a détaché de l'ensemble de la séquence un segment qui puisse raisonnablement être considéré comme un gène, il faut identifier le type de protéine codée. Le biologiste ne connaît pas nécessairement la protéine a priori, mais il peut analyser certains motifs dans sa structure afin d'en interpréter la fonction biologique, et éventuellement pour établir une parenté avec d'autres protéines connues.

1.1.3 Rôle de la modélisation markovienne

Chacune des étapes de l'annotation d'une chaîne d'ADN est longue et fastidieuse. Il faut un temps et un équipement importants pour décoder un génome, ce qui explique par exemple pourquoi l'analyse du génome humain, qui comprend environ 3 milliards de bases, a été répartie entre pays et entre laboratoires, afin d'utiliser au mieux les ressources humaines et informatiques disponibles.

Le laboratoire LSG espère accélérer l'analyse de séquences d'ADN au moyen de la modélisation par chaînes de Markov. L'idée est la suivante : au lieu de fouiller une séquence dans son intégralité pour trouver des gènes, le laboratoire suppose qu'une analyse markovienne des données permettrait d'obtenir une approximation exploitable de la structure des séquences. Celle-ci pourrait ainsi permettre de préciser les zones dans lesquelles se trouvent les gènes, ce qui diminuerait sensiblement la complexité informatique des recherches ultérieures.

De plus, à l'inverse d'une démarche purement biologique qui ne peut s'appliquer qu'à une espèce à la fois, l'analyse markovienne s'utilise simultanément sur un ensemble de séquences. Ceci permet d'établir des similitudes et des correspondances entre les séquences et les gènes. En outre, et c'est une des questions auxquelles s'attachera de répondre ce mémoire, il est essentiel de savoir si la connaissance d'un ensemble de génomes peut être exploitée dans l'analyse d'une nouvelle séquence.

1.1.4 Principe de la modélisation

Avant d'être traitée, une séquence d'ADN est un ensemble de données dont la structure est mal connue. On sait qu'elle peut être découpée selon les classifications énoncées plus haut, en régions codantes et non-codantes, en gènes et en copies de gènes, selon des fonction universelles ou spécifiques, mais on ne dispose a priori d'aucune identification de ces régions.

Une méthode statistique robuste telle que la modélisation par chaîne de Markov semble donc toute choisie. En outre, le recours aux chaînes de Markov cachées (Hidden Markov Models, HMM) est nécessaire, au vu de l'inhomogénéité des séquences d'ADN : comme nous l'avons dit, celles-ci varient par leur longueur et par la fonction des segments qui les composent.

Les HMM permettent de modéliser le découpage d'une séquence d'ADN en régimes statistiques, comme nous l'avons déjà signalé. L'objet final de leur utilisation

serait ici d'établir des correspondances entre les propriétés statistiques des régimes d'un modèle HMM et les caractéristiques biologiques des régions qui composent la séquence.

Pour effectuer ce travail, le laboratoire LSG a mis à notre disposition les génomes de onze espèces de bactéries, ainsi que le logiciel R'HOM (Research of HOMogenous regions in DNA sequences, un programme développé par le laboratoire LSG) qui permet de traiter ces données. Plus précisément R'HOM utilise l'algorithme récursif EM (Expectation Maximisation, ou maximisation de l'espérance) afin d'analyser la transition entre les bases d'une séquence, pour déterminer ensuite un partage de la séquence entre régimes homogènes. Nous reviendrons en détails sur le fonctionnement de R'HOM, ainsi que sur les fondement mathématiques des HMM et de l'algorithme EM.

1.2 Objectif

1.2.1 Principe

La quantité de bases qui composent les séquences d'ADN autorise des modélisations statistiques. Le problème biologique peut alors être transposé au langage statistique : les similitudes dans l'expression biologique des séquences d'ADN se traduisent-elles par une proximité statistique entre ces mêmes séquences ?

Le logiciel R'HOM est un outil essentiel pour tenter de répondre à cette question : il associe à la structure biologique d'une séquence un ensemble de caractéristiques statistiques sous la forme de matrices. Plus précisément, R'HOM examine les probabilités de transition entre les bases d'une séquence ; à partir de cette analyse, le logiciel découpe la séquence en régions homogènes : chaque régions est caractérisée par la matrice des probabilités de passage entre bases. On parle alors de **régimes**. Nous verrons en détails le fonctionnement de R'HOM, notamment le recours aux modèles de chaînes de Markov cachées (HMM) qui permettent de distinguer des régimes différents.

Une fois obtenues les matrices de transition associées à chaque régime d'un modèle, on peut utiliser des méthodes d'analyse des données, afin de cerner les similitudes entre régimes et espèces, et on pourra en particulier illustrer graphiquement certains résultats.

Nous obtiendrons donc un ensemble de graphes sur lesquels seront disposés des points associés aux matrices de passage des régimes définis par les modèles utilisés.

1.2.2 Mise en oeuvre

Pour assurer le bien fondé et la cohérence de notre analyse, il est nécessaire de savoir s'il existe effectivement des caractéristiques biologiques dans une séquence d'ADN qui se traduisent par des spécificités statistiques dans les segments qui lui sont associés. Puisque notre objectif est de représenter graphiquement les différences entre régions, il faut savoir comment se répartissent les régimes biologiques.

Pour cela, nous avons utilisé des séquences annotées, sur lesquelles figure le découpage des segments par rapport à des critères biologiques ; nous avons retenu trois critères : les zones **non-codantes**, et les parties **codantes positives** et **négatives**.

Nous avons alors appliqué un modèle de chaînes de Markov simples à ces données, afin de connaître les probabilités de transition des régimes biologiques. Ensuite, nous avons cherché à représenter graphiquement les distances séparant les matrices de transition. Nous avons donc utilisé une distance convenable entre matrices de transition, la Distance en Variation Totale (DVT), puis nous avons projeté dans le plan les distances entre matrices, à l'aide de la procédure MDS de SAS.

Nous avons alors pu identifier les critères statistico-biologiques de proximité entre régimes et espèces. Il est dès lors possible d'appliquer le même traitement à des séquences non annotées, en utilisant cette fois-ci des chaînes cachées à l'aide de R'HOM. Nous avons obtenu des matrices de transition purement statistiques, dont on a pu alors interpréter la disposition. Ensuite, nous avons identifié biologiquement les régimes lorsque leurs effectifs le permettaient, ce qui permet d'apprécier la qualité de la démarche, et notamment les possibilités d'utilisation en temps que méthode de prédiction. Enfin nous avons confronté directement les représentations des régimes biologiques et statistiques, ce qui nous a conduit à formuler un certain nombre de nouvelles hypothèses.

1.3 Données

1.3.1 Information disponible

Les banques de données génomiques sont en libre accès sur Internet. Le National Center for Biotechnology Information [3] (NCBI) centralise une grande masse de ces données : pour chaque espèce dont le génome a été annoté et séquencé, il entretient plusieurs bases de données, notamment les génomes "bruts" des espèces, et les génomes annotés.

Comme notre approche consiste en une analyse statistique des séquences brutes (en tant que suite de lettres a,c,g,t), des séquences annotées (en tant que suite de régions codantes ou non), nous avons eu à la fois besoin des génomes bruts et des génomes annotés. Afin de pouvoir faire la comparaison entre ces deux approches, nous avons dû nous assurer que l'information de niveau séquence et l'information de niveau annotation étaient cohérentes, c'est-à-dire qu'elles provenaient de *la même base de données.*

Or comme on l'a précisé, les bactéries ont un seul chromosome *circulaire* ; il n'existe donc pas à proprement parler de "début" ni de "fin" au génome d'une bactérie. Avant de comparer les données issues du séquençage du génome d'une bactérie, et celles issues de l'annotation de ce génome, on doit donc s'assurer qu'annotation et séquençage ont été réalisés à partir du même point de départ sur le génome. Parmi les formats de données du NCBI, le format GenBank nous a donc paru tout à fait approprié à notre étude : les fichiers GenBank sont des fichiers texte contenant à la fois les informations issues du séquençage et celles issues de l'annotation (un exemple d'extrait de fichier GenBank est donné en annexe A.1).

[3] `http://www.ncbi.nlm.nih.gov/`

1.3.2 Bactéries choisies

Nous avons étudié des séquences d'ADN tirées des génomes de 11 bactéries, dont voici une classification par espèce :

Gram +	Bacilles	bacillus subtilis
	mycotuberculium	mycobacterium tuberculosis
Gram -	chlamydia	chlamydia muridarum
	entero	escherichia coli
		haemophilus influenzae
	gonocoques	neisseria meningitidis
	pseudomonades	helicobacter pylori
		campylobacter jejuni
	spirochetes	borrelia burgdoreferi
Sans paroi		mycoplasma genitalium
		mycoplasma pneumoniae

Tableau 2: Les 11 bactéries étudiées

La longueur de ces séquences est de l'ordre du million de bases.

Chapitre 2

Analyse des données biologiques

Dans ce chapitre, on s'intéresse à l'exploitation statistique des informations d'annotation sur les génomes de nos bactéries. Les bases de données du NCBI permettent notamment de faire la distinction, au sein des génomes de bactéries, entre les parties codantes et non codantes du génome, et, à l'intérieur des segments codants, entre le codant positif et le codant négatif.

L'objectif est d'étudier les ressemblances statistiques entre ce que nous appellerons les *régimes biologiques* des génomes des bactéries que nous étudions, c'est-à-dire les plages non codante, codante positive et codante négative.

2.1 Exploitation des données

2.1.1 Données utilisées

Les fichiers GenBank du NCBI contiennent la localisation des gènes repérés par les biologistes, ainsi que toutes les informations disponibles sur ces gènes (sens de lecture du gène, traduction en acides aminés, description de la protéine pour laquelle il code, etc).

Dans un premier temps, nous avons écrit un script en langage BioPerl[1] qui sépare les portions de séquences (suites de bases) associées à chacun des régimes biologiques. Nous avons ainsi obtenu trois séquences d'ADN pour chacune de nos 11 bactéries.

Afin d'étudier le lien statistique entre ces 33 séquences, nous avons choisi de résumer l'information qu'elles contenaient à l'aide de modèles de Markov simples.

2.1.2 Chaînes de Markov simples

Une chaîne de Markov est une suite $(X_t)_{t\geq 0}$ de variables aléatoires à valeurs dans un ensemble $\mathcal{E}$ fini [2] qui vérifie la propriété suivante :

$$\mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}, ..., X_0 = x_0) = \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1})$$

[1] BioPerl est un ensemble de classes écrites en langage objet Perl et adaptées au traitement des bases de données biologiques.

[2] On peut aussi prendre $\mathcal{E}$ infini, mais ce ne sera jamais notre cas.

pour $t \geq 1$ et $x_0, ..., x_t$ dans $\mathcal{E}$. On peut alors définir la suite $(\pi_t)_{t\geq 0}$ des matrices de transition de X qui sont les matrices de terme général :

$$\pi_t(a,b) = \mathbb{P}(X_t = b | X_{t-1} = a)$$

Dans tout ce travail, nous ne considérons que des chaînes de Markov homogènes, c'est à dire pour lesquelles la suite des matrices de transition est constante. On notera alors π l'unique **matrice de transition** de la chaîne X.

Si la loi de X_{t-1} est donnée par un vecteur colonne L, alors la loi de X_t est donnée par ${}^t\pi L$. Comme π est une matrice stochastique, elle admet 1 comme valeur propre et c'est donc aussi le cas de ${}^t\pi$. Un vecteur propre normalisé de ${}^t\pi$ associé à la valeur propre 1 représente une **distribution stationnaire** : si la loi de X_0 est représentée par un tel vecteur, alors les lois de tous les X_t le seront aussi.

Voici une simulation [3] avec $\mathcal{E} = \{a, t, c, g\}$, et $\pi = \begin{pmatrix} 0.50 & 0.24 & 0.24 & 0.02 \\ 0.01 & 0.01 & 0.01 & 0.97 \\ 0.01 & 0.01 & 0.01 & 0.97 \\ 0.97 & 0.01 & 0.01 & 0.01 \end{pmatrix}$:

atgatgaaatgaaatgaatgaaaatgaaacgaatgaaaaatgaaa
atgatgaaatgaaatgaatgaaaatgaaacgaatgaaaaatgaaa
aacgacgacgacgatgaatgacggcgaatgatgaatgacgatgac
gatgaaaacgacgacgatgatgacgatgaaacgacgatgaaaatg
aatgacgtgaaaacgatgaaacgtgaatgaattgaatgatgaacg
atgacgacgaaaaaaaaatgatgaacgatgaaaatgaacgaaaaa
tgaaatgaaaacgatgaacgatgaatgatgaaatgatgatgacga
acgatgaaccgaacgatgaaaaaacgaaacacgatgaaacgaaa...

On obtient une chaîne possédant des caractéristiques facilement reconnaissables : du fait que le A est "bouclé sur lui-même", on trouve des plages de A relativement longues ; ces plages sont presque tout le temps suivies de doublets CG et de doublets TG.

Cette simulation donne une idée du genre de motifs qui peuvent être modélisés par une chaîne de Markov simple.

2.1.3 Chaînes de Markov avec mémoire

On peut généraliser légèrement le modèle en introduisant une notion de mémoire : une chaîne de Markov de mémoire k (modèle Mk) est une suite de variables aléatoires qui vérifie la propriété :

$$\mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}, ..., X_0 = x_0) = \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}, ..., X_{t-k} = x_{t-k})$$

pour $n \geq k$ et $x_0, ..., x_t$ dans $\mathcal{E}$.

[3] Cette simulation, ainsi que celle que l'on trouvera plus loin dans la partie sur les chaînes de Markov cachées, a été réalisée à l'aide d'un programme Java que nous avons écrit.

Comme pour les chaînes simples, on peut définir les matrices de transition du processus que l'on supposera à nouveau indépendantes de t, de sorte que l'on parlera de "la" matrice de transition de X. C'est la matrice de terme général :

$$\pi(a_1, ..., a_k, a_{k+1}) = \mathbb{P}(X_t = a_{k+1} | X_{t-1} = a_k, ..., X_{t-k} = a_1)$$

En fait, les chaînes de Markov avec mémoire peuvent être vues comme des chaînes de Markov simples au prix d'une modification de l'espace des états. Il suffit de poser pour tout $t \geq k$

$$Z_t = (X_t, ..., X_{t-k+1}).$$

Alors Z est une chaîne de Markov simple à valeur dans $\mathcal{E}^k$.

Ce changement de point de vue affecte les matrices de transition. La matrices de transition de Z est carrée et s'obtient à partir de la matrice de transition (rectangulaire) de X en ajoutant beaucoup de zéros. Prenons l'exemple d'une chaîne de mémoire 2 à valeurs dans $\{a, t, c, g\}$. La chaîne Z n'a que quatre transitions possibles à partir d'un doublet donné (au lieu de seize a priori) car le nouveau doublet doit commencer par la dernière lettre du précédent. Ainsi par exemple, les seules transitions possibles à partir de ca sont $ca \to aa$, $ca \to at$, $ca \to ac$ et $ca \to ag$. Il y aura donc des zéros dans la matrice de transition de Z aux endroits qui correspondent à des transitions impossibles.

Montrons ce qui se passe sur un ensemble $\mathcal{E}$ à trois éléments (car les matrices 16×16 ne tiennent pas dans la page). Si la matrice de transition de X est de la forme :

$$\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix}$$

alors la matrice de transition de Y sera :

$$\begin{pmatrix} \times & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & \times & \times \\ \times & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & \times & \times \\ \times & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & \times & \times \end{pmatrix}$$

Pour estimer la matrice de transition d'une chaîne de Markov de mémoire quelconque, on calcule la fréquence d'apparition de chaque transition :

$$\hat{\pi}(a_1, ..., a_k, a_{k+1}) = \frac{\sum_{t=k+1}^{T} 1_{\{X_t=a_{k+1},...,X_{t-k}=a_1\}}}{\sum_{t=k+1}^{T} 1_{\{X_{t-1}=a_k,...,X_{t-k}=a_1\}}}$$

où T est la longueur de la chaîne étudiée.

Dans notre cas, le calcul effectif des fréquences d'apparition des transitions dans les séquences constituées de régimes biologiques est un peu plus compliqué : en effet, ces séquences ne sont pas connexes, car les régions codantes positives, codantes négatives, et non codantes alternent tout au long du génome. Pour les bactéries que nous avons étudiées, on dénombre de quelques centaines à quelques milliers de gènes, séparés par autant de régions non codantes.

Pour le calcul des fréquences d'apparition des différentes transitions, on doit donc faire attention à ne prendre en compte que les transitions au sein d'une partie connexe d'une séquence. Nous avons utilisé le logiciel Coton, développé par Pierre-Yves Bourguignon, pour faire ce calcul.

2.1.4 Distance en variation totale

Pour mener à bien notre étude, nous avons besoin de pouvoir comparer des régimes markoviens entre eux. Pour cela nous allons définir une distance entre chaînes de Markov. Nous utiliserons la distance en variation totale, qui est une distance classique entre lois de probabilités.

De manière générale, si $\mathbb{P}^{(1)}$ et $\mathbb{P}^{(2)}$ sont des probabilités sur un ensemble fini $\mathcal{X}$, alors la distance en variation totale entre $\mathbb{P}^{(1)}$ et $\mathbb{P}^{(2)}$ est définie par :

$$dvt(\mathbb{P}^{(1)}, \mathbb{P}^{(2)}) = \sum_{x \in \mathcal{X}} |\mathbb{P}^{(1)}(x) - \mathbb{P}^{(2)}(x)|$$

Considérons maintenant deux chaînes de Markov $X^{(1)}$ et $X^{(2)}$ de mémoire k. Pour pouvoir calculer leur distance en variation totale, il faut leur associer des lois de probabilités sur un certain ensemble fini. Il est naturel de choisir la loi des $(k+1)$-uplets de lettres successives, car celle-ci caractérise entièrement la loi d'une chaîne de Markov de mémoire k.

On pose donc $\mathcal{X} = \mathcal{E}^{k+1}$ et pour $a_1, ..., a_{k+1} \in \mathcal{E}$:

$$\mathbb{P}^{(1)}(a_1, ..., a_{k+1}) = \mathbb{P}(X_t^{(1)} = a_{k+1}, ..., X_{t-k}^{(1)} = a_1)$$

$$\mathbb{P}^{(2)} = \mathbb{P}(X_t^{(2)} = a_{k+1}, ..., X_{t-k}^{(2)} = a_1)$$

$\mathbb{P}^{(1)}$ et $\mathbb{P}^{(2)}$ peuvent se calculer à l'aide des matrices de transition et des distributions stationnaires de chaque chaîne. Si on note $\pi^{(1)}$ (resp. $\pi^{(2)}$) la matrice de transition de $X^{(1)}$ (resp. $X^{(2)}$) et $\mu^{(1)}$ (resp. $\mu^{(2)}$) la loi stationnaire de $X^{(1)}$ (resp. $X^{(2)}$) alors :

$$\mathbb{P}^{(1)}(a_1, ..., a_{k+1}) = \mu^{(1)}(a_1, ..., a_k)\pi^{(1)}(a_1, ..., a_{k+1})$$

Donc on a finalement :

$$dvt(X^{(1)}, X^{(2)}) = \sum_{(a_1,...,a_{k+1}) \in \mathcal{E}^{k+1}} |\mu^{(1)}(a_1, ..., a_k)\pi^{(1)}(a_1, ..., a_{k+1}) - \mu^{(2)}(a_1, ..., a_k)\pi^{(2)}(a_1, ..., a_{k+1})|$$

2.1.5 *Multidimensional scaling*

Maintenant que nous avons défini une distance entre régimes markoviens, nous avons besoin d'une méthode de représentation qui nous aide à comprendre comment nos points sont organisés par cette distance. Nous avons utilisé le **multidimensional scaling** (**MDS**) qui est la technique classique pour traiter ce genre de problèmes. Ce terme désigne en fait un ensemble de techniques, mais nous n'aurons besoin que de l'une d'entre elles, qui est assez élémentaire et que nous allons décrire maintenant.

Celle-ci consiste à plonger l'ensemble I des points à représenter muni de sa distance δ (dans notre cas, la distance en variation totale) dans un espace $\mathbb{R}^p$ muni de la distance euclidienne canonique, que nous noterons d. Ce plongement doit bien sûr être une isométrie.

Supposons que l'ensemble I contienne n points. Pour i et j dans I, on pose $h_{ij} = \delta^2(i,j)$. On définit ensuite une matrice B de dimension $n \times n$ par la formule :

$$b_{ij} = -\frac{1}{2}(h_{ij} - h_{i.} - h_{.j} + h_{..})$$

où les points signifient qu'on a pris une moyenne sur l'ensemble des valeurs possibles de l'indice qu'ils remplacent.

B est alors une matrice définie positive et on peut donc trouver une matrice X de dimension $n \times n$ telle que $B = X^t X$. Si note x_i le vecteur correspondant à la i^{eme} colonne de X, alors on a pour i et j dans I :

$$d^2(x_i, x_j) = b_{ii} + b_{jj} - 2b_{ij} = h_{ij} = \delta^2(i,j)$$

On a bien trouvé n points de $\mathbb{R}^p$ (avec $p = n$) dont les distance mutuelles sont exactement les mêmes que celles de points de I.

L'intérêt de placer les points dans un espace euclidien est que l'on peut maintenant les projeter dans un plan et en faire une représentation graphique. Bien sûr, on ne choisit pas le plan n'importe comment : on utilise une ACP de manière à ce que les points soit aussi étalés que possible.

Pour réaliser l'ensemble de ces opérations, nous avons utilisé la procédure MDS de SAS.

2.2 Résultats

Les graphes que nous allons analyser ici sont des projections planes des points que la procédure MDS associe à chaque régime. Chaque point correspond à un couple (régime, bactérie).

Les points sont étiquetés en utilisant la notation xy_Rb, où xy est l'acronyme de la bactérie, Rb le régime biologique, Pb désignant le régime codant positif, Mb le codant négatif, et Ob le non-codant.

acronyme	bactérie	acronyme	bactérie
bb	borrelia burgdoreferi	hp	helicobacter pylori
bs	bacillus subtilis	mg	mycoplasma genitalium
cj	campylobacter jejuni	mp	mycoplasma pneumoniae
ct	chlamydia muridarum	mt	mycobacterium tuberculosis
ec	escherichia coli	nm	neisseria meningitidis
hi	haemophilus influenzae		

2.2.1 *Groupement par espèce*

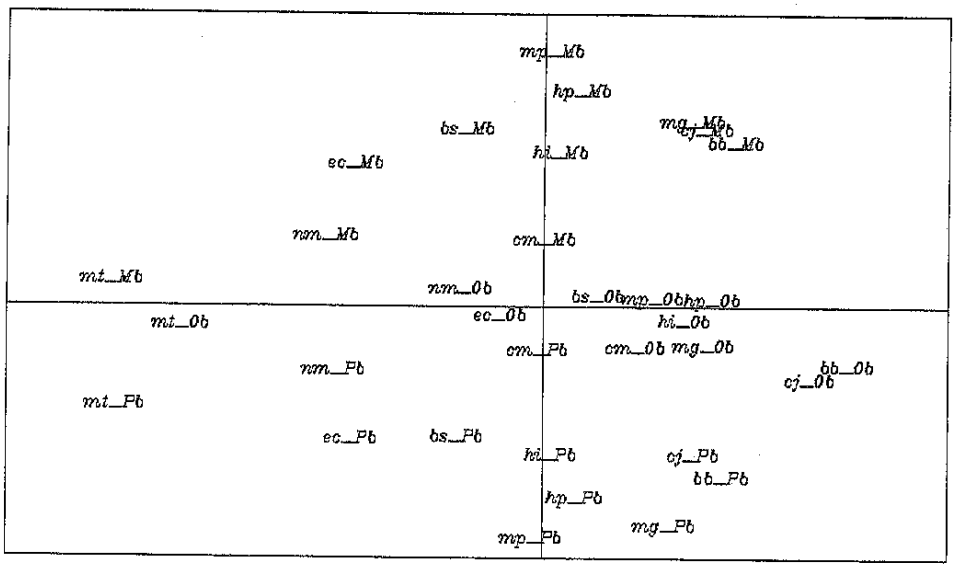

Figure 2.1: Groupement par espèce

Ce premier graphe est le plus général. Il représente les points associés aux trois régimes des 11 bactéries étudiées.

La première caractéristique de ce graphe est sa relative symétrie par rapport à l'axe des abscisses (axe 1). En regardant de plus près, on constate que pour chaque bactérie, on trouve systématiquement le régime codant positif (Pb) sous l'axe 1, le codant négatif au dessus, et le régime non-codant proche de l'axe. En d'autres termes, l'axe des ordonnées (axe 2) permet de différencier les régimes.

Ensuite, on observe une certaine proximité entre régimes d'une même espèce. Si l'on ne considère plus la distance induite par l'effet régime associé à l'axe 2, l'axe 1 distingue très bien les bactéries : les trois points associés à une même espèce ont leurs abscisses très proches.

En revanche, l'étude de ce graphe à la lumière de la classification des espèces fournit peu d'informations significatives. Les espèces sans parois mycoplasma genitalium (mg) et mycoplasma pneumoniae (mp) figurent parmi les bactéries dont les régimes sont les plus distants entre eux. Cependant, la différence avec d'autres espèces est loin d'être significative ; en particulier helicobacter pylori (hp) est une pseunomonade (famille Gram-) dont les régimes apparaissent aussi espacés.

Les familles Gram- et Gram+ ne sont pas identifiables, et les sous-groupes qu'elles contiennent ne le sont pas non plus. Ainsi, escherichia coli (ec) et haemophilus influenzae (hi) sont deux entero (Gram-) ; elles sont plus proches biologiquement, mais sur le graphe, leur proximité ne les distingue pas des autres bactéries. Bacillus subtilis est d'ailleurs plus proche de chacune de ces deux bactéries, alors qu'elle est une Gram+. La situation est la même pour les pseudomonades (Gram-) helicobacter pylori (hp) et campylobacter jejuni (cj)

Après l'étude de ce premier graphe, nous avons d'abord cherché à supprimer l'information la moins utile, c'est-à-dire la distance séparant les espèces de bactéries, puisqu'elle fournit peu d'interprétations. Pour cela, nous avons centré pour chaque bactérie le nuage de trois points qui lui correspond. Nous avons naturellement pondéré les points par la longueur des régimes qu'ils représentent. Cette démarche permet de privilégier l'effet régime, et donc de mieux analyser la distance entre les points associés aux régimes des bactéries.

2.2.2 Groupement par régimes biologiques

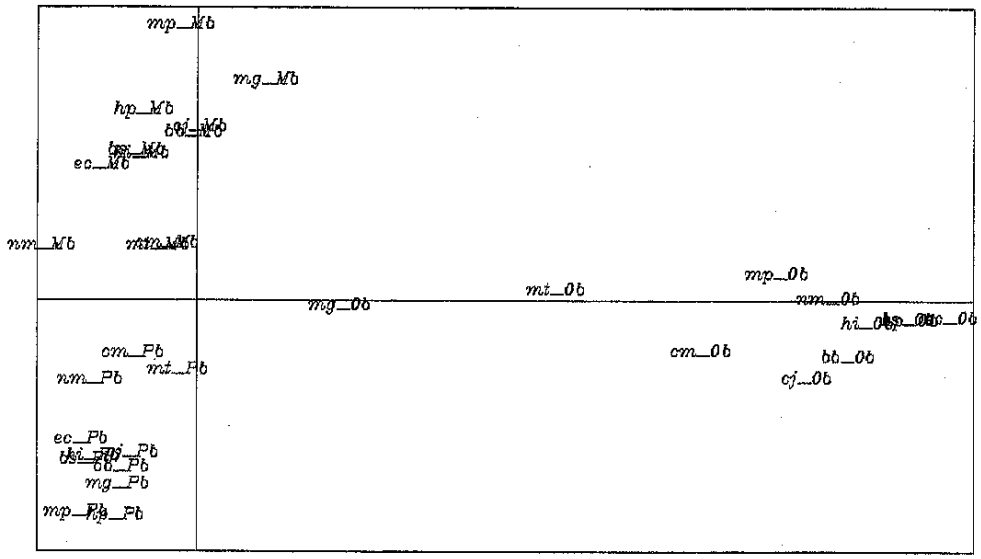

Figure 2.2: Groupement par régimes biologiques

On distingue clairement sur ce deuxième graphe trois nuages de points :
- à droite, proche de l'axe 1, on trouve pour chaque espèce le point associé au régime non codant.
- le régime codant positif se situe dans le cadrant en bas à gauche.
- en haut à gauche, on observe le codant négatif.

D'autres précisions peuvent être apportées par rapport à l'analyse du graphe non-centré, dans la mesure où ce second graphe permet d'analyser les points par régime. Les remarques précédentes semblent désormais valables pour certains régimes : les espèces sans parois mycoplasma genitalium (mg) et mycoplasma pneumoniae (mp) se distinguent sensiblement dans les régimes non-codant et codant négatif, alors que leurs points se confondent parmi les autres dans le codant positif. En outre,

dans le cas de mycoplasma pneumoniae, on peut penser que ses régimes sont non seulement très éloignés, mais aussi tous décalés vers le haut par rapport au reste des points, ce qui explique le manque de distinction du régime positif. Néanmoins, cette remarque n'est pas valable pour mycoplasma genitalium, donc elle ne peut être raisonnablement étendue.

Comme sur le graphe 1, on n'observe pas de distinction globale entre familles d'espèces. Cependant, là où le graphe 1 ne permettait pas d'observer de proximité entre espèces proches, on distingue au contraire sur cette deuxième figure plusieurs dispositions qui méritent d'être mentionnées. D'une part, les deux Gram+ bacillus subtilis (bs) et mycobacterium tuberculosis (mt) ont leurs régimes non-codants proches. D'autre part, les pseudomonades (Gram-) helicobacter pylori (hp) et campylobacter jejuni (cj) ont leurs régimes codants négatifs très proches, quoique cette proximité ne les distingue pas de la spirochète borrelia burgdoreferi (bb), qui est systématiquement plus proche de pylori et jejuni, notamment pour le codant négatif.

Chapitre 3

Analyse des données statistiques

3.1 Traitement des données

3.1.1 Chaînes de Markov cachées

Pour résumer l'idée des chaînes de Markov cachées en une phrase, on peut dire qu'il s'agit de chaînes dont la matrice de transition change de valeur suivant une chaîne de Markov inobservable.

Plus formellement, un modèle de chaîne de Markov cachée de type $M1Mk$ à n régimes est la donnée de deux suites de variables aléatoires $(S_t)_{t\geq 0}$ (la chaîne cachée) et $(X_t)_{t\geq 0}$ (la chaîne observée), vérifiant les conditions suivantes :

(i) S est une chaîne de Markov de mémoire 1 sur l'ensemble $\{1, ..., n\}$.

(ii) X est une suite à valeurs dans un ensemble fini $\mathcal{E}$ (pour nous ce sera $\{a, t, c, g\}$).

(iii) $\mathbb{P}(X_t = x_t | S_t = s_t, X_{t-1} = x_{t-1}, S_{t-1} = s_{t-1}, ..., X_0 = x_0, S_0 = s_0)$
$= \mathbb{P}(X_t = x_t | S_t = s_t, X_{t-1} = x_{t-1}, ..., X_{t-k} = x_{t-k})$
pour $t \geq k$, $x_0, ..., x_t$ dans $\mathcal{E}$ et $s_0, ..., s_t$ dans $\{1, ..., n\}$.

Dans un tel modèle, on définit deux matrices de transition. Comme dans le cas des modèles de Markov simples, on fait l'hypothèse que les termes généraux de ces matrices sont indépendants de t, faute de quoi il faudrait introduire des suites de matrices de transition.

On note π_s la **matrice de transition de la chaîne cachée** S :

$$\pi_s(u, v) = \mathbb{P}(S_t = v | S_{t-1} = u)$$

$$u, v \in \{1, ..., n\}$$

et on note π_o la **matrice de transition de la chaîne observée** X :

$$\pi_o(u, a_1, ..., a_{k+1}) = \mathbb{P}(X_t = a_{k+1} | S_t = u, X_{t-1} = a_k, ..., X_{t-k} = a_1)$$

$$u \in \{1, ..., n\}\ a_1, ..., a_{k+1} \in \mathcal{E}$$

La matrice π_o peut être vue comme un ensemble de n matrices de transition : $\pi_o(1), ..., \pi_o(n)$. La transition qui a lieu à un "instant" t dépend de la valeur de S_t.

Si $S_t = u$ alors la transition est régie par la matrice $\pi_o(u)$.

Voici une simulation de modèle $M1M1$ sur $\mathcal{E} = \{a, t, c, g\}$ avec deux états cachés. On a pris π_s proche de la matrice identité : $\pi_s = \begin{pmatrix} 0,99 & 0,01 \\ 0,05 & 0,95 \end{pmatrix}$, de manière à ce que les changements de régimes ne soient pas trop fréquents. Cela permet d'obtenir des régimes relativement longs, à l'instar des régimes biologiques. Pour régir les transitions du régime 1, on reprend la matrice $\pi_o(1) = \begin{pmatrix} 0.50 & 0.24 & 0.24 & 0.02 \\ 0.01 & 0.01 & 0.01 & 0.97 \\ 0.01 & 0.01 & 0.01 & 0.97 \\ 0.97 & 0.01 & 0.01 & 0.01 \end{pmatrix}$ qui avait servi à simuler une chaîne de Markov simple dans la partie précédente. Pour le régime 2, on utilise $\pi_o(2) = \begin{pmatrix} 0,01 & 0,97 & 0,01 & 0,01 \\ 0,01 & 0,01 & 0,97 & 0,01 \\ 0,01 & 0,01 & 0,01 & 0,97 \\ 0,97 & 0,01 & 0,01 & 0,01 \end{pmatrix}$. De manière à ce que les changements de régime soient visibles, on utilise des lettres majuscules quand on se trouve dans le régime 1 et des minuscules quand on se trouve dans le régime 2 :

ATGATcatcgatcgatcgatatcGATGGATGAAACGAAAAttcgatcGATGAAAT
GAATGAGAAAATGAAAAAAACGATGATGATGCGACGAAACGAAAA
AGAACCGACGAAAATGATGATGATGAACAACGACGACGATGAAAC
GAACGATATGATGATGAATGACGAAATGAAAAATGATGATGACGA
TGATGatcgattcgatttcgatctatcgAATGAAAAAAACGATGAAACGATGAT
TGACGAACGATcatccgatcgatatcgatcgACGACGAACGATGACGACGAAA
GAACGACGACGACGACGAAACGATGAAATGATGACGATGACGAC...

On reconnaît les traits caractéristiques du régime 1 avec ses "...AAACG" et ses "...AAATG". Le régime 2, quant à lui, est constitué de suites de "atcg" avec quelques perturbations. On peut imaginer que quelqu'un qui lirait cette suite sans pouvoir distinguer les lettres majuscules des lettres minuscules arriverait tout de même à retrouver approximativement la position des régimes, car ceux-ci ont des allures très différentes. Il pourrait ensuite estimer les matrices de transition de chaque régime, ainsi que celle de la chaîne cachée.

Nous allons maintenant voir comment cette démarche approximative peut être systématisée à l'aide de l'algorithme EM qui est implémenté dans le logiciel R'HOM.

3.1.2 Principe de l'algorithme EM

L'estimation des chaînes de Markov cachées ne se ramène pas à un simple calcul de fréquence d'apparition des transitions comme c'est le cas pour les chaînes de Markov simples. En effet, pour dénombrer les transitions qui ont lieu dans un régime donné, il faut commencer par savoir où se trouve ce régime dans la chaîne. Cette difficulté est surmontée à l'aide de l'algorithme EM que nous allons exposer maintenant.

L'algorithme EM est un algorithme itératif. Son nom provient du fait qu'à chaque itération, il y a alternance d'une phase (E) de calcul d'espérance et d'une phase (M) de maximisation. Cet algorithme est très général et peut être utilisé dès qu'on est en présence de données incomplètes. Nous allons en donner le principe général et nous

verrons ensuite comment l'appliquer au cas des chaînes de Markov cachées.

On considère donc un modèle avec données incomplètes : le modèle fournit la densité d'une variable Y (dite complète) en fonction d'un paramètre θ mais on observe uniquement la variable incomplète X, image de Y par une fonction non injective [1].

L'objectif de l'algorithme EM est de trouver l'estimateur du maximum de vraisemblance de θ. Un calcul direct est théoriquement possible. Celui-ci consiste dans un premier temps à calculer la densité de X en fonction de θ à partir de celle de Y. Ensuite on maximise cette expression en la variable θ pour la valeur de X qu'on a observée. On obtient ainsi directement l'estimateur du maximum de vraisemblance. Mais en pratique, l'expression analytique de la densité de X est souvent très complexe par rapport à celle de la densité de Y et cette méthode pose donc de gros problèmes de calcul.

Dans l'algorithme EM, on se tourne vers la densité de Y, en principe plus facile à utiliser dans les calculs. Le problème, c'est que Y n'est pas observé donc cette densité ne peut pas être utilisée directement comme une vraisemblance. Si on observait Y, on calculerait simplement la vraisemblance de cette observation en fonction de θ. Comme on n'observe pas Y, cette vraisemblance devient une variable aléatoire et la maximisation ne peut pas se faire directement.

L'idée est donc de faire précéder la maximisation par une phase d'estimation de la vraisemblance de Y. En fait dans l'algorithme EM, on estime la log-vraisemblance. Cette estimation se fait en calculant l'espérance de la log-vraisemblance de Y conditionnellement à tout ce qu'on connaît, c'est à dire X. Mais pour calculer cette espérance conditionnelle, il faut connaître la loi de Y et donc θ. On est donc dans un cercle vicieux, dont on se sort par l'emploi d'une méthode itérative.

On part donc d'une valeur quelconque de θ que l'on notera $\theta^{(0)}$. On peut alors calculer l'espérance de la log-vraisemblance de Y conditionnellement à la valeur de X observée et sous l'hypothèse que Y suit la loi de paramètre $\theta^{(0)}$. Si on note $f_\theta(y)$ la densité de la loi de paramètre θ et $L(\theta, Y) = log(f_\theta(Y))$ la log-vraisemblance en θ, l'estimation de la log-vraisemblance consiste à calculer la fonction Q définie par :

$$Q(\theta, \theta^{(0)}) = \mathbb{E}_{\theta^{(0)}}(L(\theta, Y)|X = x)$$

Ce calcul constitue la phase (E). On calcule ensuite une nouvelle estimation de θ en maximisant cette log-vraisemblance estimée par rapport à la variable θ. C'est la phase (M). Ensuite on recommence cette alternance de phase (E) et phase (M) jusqu'à avoir atteint la précision souhaitée sur θ.

En résumé, la suite des $\theta^{(m)}$ est donnée par la formule

$$\theta^{(m+1)} = Argmax\ Q(\theta, \theta^{(m)})$$

où

$$Q(\theta, \theta^{(m)}) = \mathbb{E}_{\theta^{(m)}}(L(\theta, Y)|X = x)$$

[1] Par exemple, on peut imaginer un daltonien qui tire des boules dans une urne. Il y a trois couleurs, mais pour lui deux d'entre elles sont identiques (disons les deux premières). Ainsi, si n boules sont tirées et s'il y a parmi elles A boules de la première couleur, B boules de la deuxième couleur et C boules de la troisième couleur, il n'observera que $X = (A + B, C)$ et non $Y = (A, B, C)$.

La suite ainsi construite converge vers l'estimateur du maximum de vraisemblance de θ et on peut montrer qu'à chaque itération, la log-vraisemblance augmente. On peut donc arrêter les itérations quand la variation de la log-vraisemblance passe en dessous d'un seuil fixé à l'avance.

3.1.3 Application d'EM aux chaînes de Markov cachées

Le cas des chaines de Markov cachées rentre bien dans le cadre des données incomplètes puisqu'on observe une chaîne X (dans notre cas, la chaîne d'ADN) mais qu'on suppose celle-ci issue d'une double chaîne $Y = (X, S)$, dont la partie cachée S est la suite des régimes.

Le paramètre θ à estimer se compose de la matrice de transition π_s de la chaîne cachée et de la matrice de transition π_0 de la chaîne observée. Le modèle spécifie donc bien la loi de Y et non celle de X.

La phase (E) de l'algorithme se ramène à calculer les lois des couples (S_t, S_{t+1}) conditionnellement à la valeur de la chaîne observée, pour la valeur en cours $\theta^{(m)}$ du paramètre. Cela est dû au caractère markovien d'ordre 1 de la suite des états cachés. On calcule donc :

$$\mathbb{P}_{\theta^{(m)}}(S_t = u, S_{t+1} = v|X) \; i = 1, ..., T-1$$

Ce calcul se fait à l'aide de la méthode de Baum-Welch (aussi appelée "forward-backwerd") que nous ne pouvons pas détailler ici. Disons simplement qu'elle consiste en un calcul de proche en proche des probabilités $\mathbb{P}(S_t = i, X_0 = x_0, ..., X_t = x_t)$ (forward) et $\mathbb{P}(S_t = i, X_t = x_t, ..., X_T = x_T)$ (backward).

La phase (M) de l'algorithme conduit aux estimateurs suivants :

$$\pi_s^{(m+1)}(u, v) = \frac{\sum_{t=1}^{T-1} \mathbb{P}_{\theta^{(m)}}(S_t = u, S_{t+1} = v|X)}{\sum_{t=1}^{T-1} \mathbb{P}_{\theta^{(m)}}(S_t = u|X)}$$

$$\pi_0^{(m+1)}(u, a_1, ..., a_{k+1}) = \frac{\sum_{t=k+1}^{T} \mathbb{P}_{\theta^{(m)}}(S_t = u|X) 1_{\{X_t = a_{k+1}, ..., X_{t-k} = a_1\}}}{\sum_{t=k+1}^{T} \mathbb{P}_{\theta^{(m)}}(S_t = u|X) 1_{\{X_{t-1} = a_k, ..., X_{t-k} = a_1\}}}$$

L'estimateur de π_s est assez intuitif. En effet, si on pouvait délimiter de façon nette le régime u, il serait naturel d'estimer $\pi_0^{(m+1)}(u, a_1, ..., a_{k+1})$ par la fréquence d'apparition de a_{k+1} après $(a_1, ..., a_k)$ dans le régime u (c'est ce qu'on fait dans le cas des chaînes de Markov simples). L'estimateur que nous utilisons fait exactement la même chose sauf qu'il tient compte du flou qui existe sur la délimitation du régime u à travers la pondération par $\mathbb{P}_{\theta^{(m)}}(S_t = u|X)$.

3.1.4 Le logiciel R'HOM

Le logiciel *R'HOM*, qui a été développé par ... , et qui est maintenent courament utilisé par les biologistes, implémente la version de l'algorithme EM que nous venons de décrire.

Il prend en entrée une chaîne d'ADN [2], une spécification de modèle (nombre de régimes et mémoire de la chaîne observée) et une valeur initiale de π_s et π_0.

En sortie, on obtient d'une part une estimation des matrices π_s et π_0 et d'autre part une estimation, pour chaque position t dans la chaîne et chaque régime u, de la probabilité $\mathbb{P}(S_t = u|X)$. Dans la pratique, il existe souvent une valeur de u pour laquelle cette probabilité est nettement plus grande que les autres de sorte qu'on peut attribuer à chaque position t un régime u sans grand rique d'erreur.

Le choix des valeurs initiales de π_s et π_0 est très important car la log-vraisemblance peut présenter des maxima locaux. Pour éviter que la suite ne converge vers l'un d'entre eux, R'HOM offre la possibilité de choisir simultanément plusieurs valeurs initiales de π_s et π_0 et de ne garder finalement que celle qui donne la log-vraisemblance la plus grande après un nombre d'itérations donné.

Nous avons extrait les séquences d'ADN des bases de données GenBank à partir d'un script BioPerl. Cependant, les données extraites n'étaient pas exploitables directement par R'HOM, car elles contenaient d'autres lettres que a,c,g, et t. En effet, il arrive que les biologistes n'aient pas pu identifier quelques bases de la séquence. Dans ce cas, la base en question est remplacée dans la séquence par une autre lettre, qui dépend de l'information dont on dispose sur la base. La nomenclature utilisée est celle de l'IUPAC (International Union of Pure and Applied Chemistry).

R	A, G	Purine
Y	C, T	Pyrimidine
W	A, T	Faible (weak) interaction de la liaison hydrogène
S	C, G	Forte (strong) interaction de la liaison hydrogène
M	A, C	Groupes NH_2 (amino) en positions identiques
K	G, T	Groupes ketones en positions identiques
B	C, G, T	Tout sauf A (premier symbole libre qui suit A)
D	A, G, T	Tout sauf C (premier symbole libre qui suit C)
H	A, C, T	Tout sauf G (premier symbole libre qui suit G)
V	A, C, G	Tout sauf T (premier symbole libre qui suit T)
N	A, C, G, T	N'importe quel (any) symbole

Figure 3.1: Nomenclature IUPAC

Concrètement, les séquences que nous avons étudiées comportaient très peu de telles lettres (au maximum une centaine sur plus d'un million de bases). Cependant, nous n'avons pas pu ignorer ce problème, car R'HOM ne fonctionne pas en présence de ces lettres. Nous avons donc écrit un programme en langage C qui transforme nos séquences en des fichiers utilisables par R'HOM, en remplaçant les bases indéterminées par une des lettres a,c,g ou t. Le code de ce programme est donné en annexe A.2.

[2] Il peut en fait être utilisé avec plusieurs alphabets, comme par exemple un alphabet d'acides aminés, ce qui permet d'analyser des protéines.

Notons que notre programme ne remplace pas une base indéterminée par n'importe quelle base : il utilise au maximum l'information disponible dans la séquence, afin de minimiser les erreurs commises. Par exemple, s'il rencontre un 'y' (ce qui signifie que la base est une pyrimidine, c'est-à-dire soit une cytosine, soit une thymine), il choisira 'c' ou 't' en fonction de la fréquence d'apparition de ces deux lettres le long de la séquence.

3.2 Résultats

Les points sont étiquetés en utilisant la notation xyn_r, où xy est l'acronyme de la bactérie, n le nombre de régimes statistiques du modèle, et r l'indice du régime (par exemple, pour le modèle à 3 régimes cachés de Campylobacter jejuni, les trois points sont nommés cj3_1, cj3_2 et cj3_3.

L'analyse biologique des sorties de MDS étudiées dans le chapitre précédent nous ont appris plusieurs choses. D'abord, on observe un effet bactérie très net, qui groupe les points associés aux régimes d'une bactérie. Ensuite, nous avons remarqué un effet régime, particulièrement visible sur le graphe centré, qui est tel que les points correspondants à un même régime se regroupent entre eux, formant trois nuages bien distincts. Enfin, nous n'avons pas constaté de net reflet de la classification biologique des bactéries.

C'est sur ces trois constats que s'est agencée notre analyse des régimes statistiques déterminés par R'HOM. Seulement, R'HOM apporte une nouvelle dimension au problème : on peut choisir le nombre de régimes du découpage d'une séquence. Ceci permet d'approfondir deux points. D'une part, nous avons voulu comprendre comment R'HOM passe d'un découpage en x régimes à un découpage à x+1 régimes. D'autre part, puisque nous disposons d'un découpage biologique en trois régimes, il est nécessaire de savoir comment les régimes de R'HOM soutiennent la comparaison avec la réalité.

3.2.1 Groupement par espèce

Nous avons débuté par une représentation des points obtenus dans un modèle à un seul régime. Les bactéries sont nettement séparées, mais la proximité des familles d'espèces n'est pas visible. Tout ceci est conforme à ce que l'étude biologique nous a appris.

En passant à un modèle à deux régimes, puis à trois, quatre et cinq régimes, on observe à chaque fois et pour chaque bactérie le remplacement d'un des points par deux nouveaux points, de telle sorte que l'ancien point est un barycentre des deux nouveaux, alors que les autres points restent pratiquement inchangés.

Donc les régimes obtenus par R'HOM ont une sorte de priorité : ils existent dans des modèles à régimes plus ou moins nombreux, et demeurent inchangés un certain nombre de fois lorsqu'on augmente le nombre de régimes.

Pour approfondir cet idée, considérons pour une bactérie donnée le graphe comportant l'ensemble des points associés obtenus par les modèles à un, deux, trois, quatre et cinq régimes.

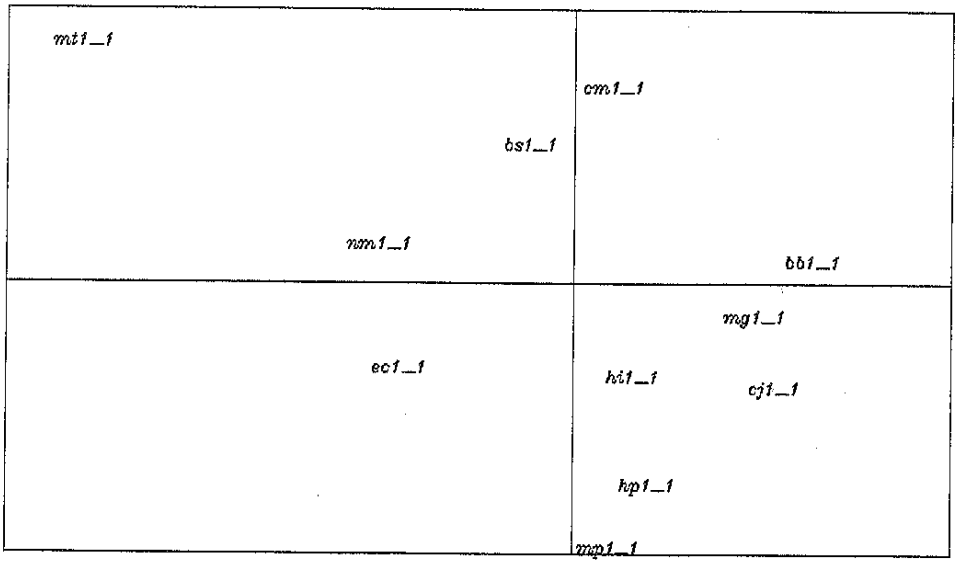

Figure 3.2: Modèle à un seul régime — Groupement par espèce

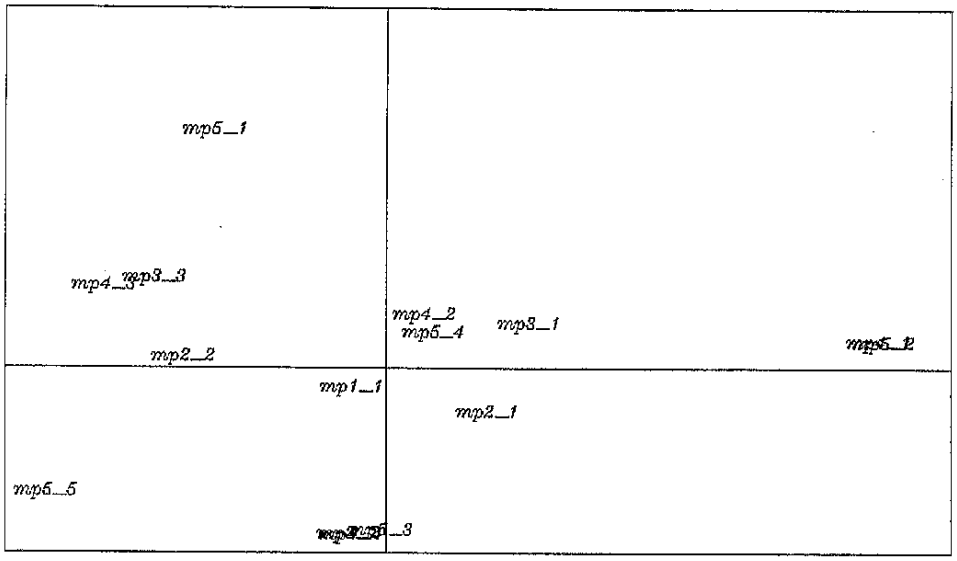

Figure 3.3: Modèles de 1 à 5 régimes — Mycobacterium tuberculosis

Sur la figure 3.3, on constate que de un à cinq régimes, la différenciation est presque parfaite : lorsque le nombre de régimes augmente, les points demeurent pratiquement inchangés sauf un, qui est dédoublé en deux nouveaux points.

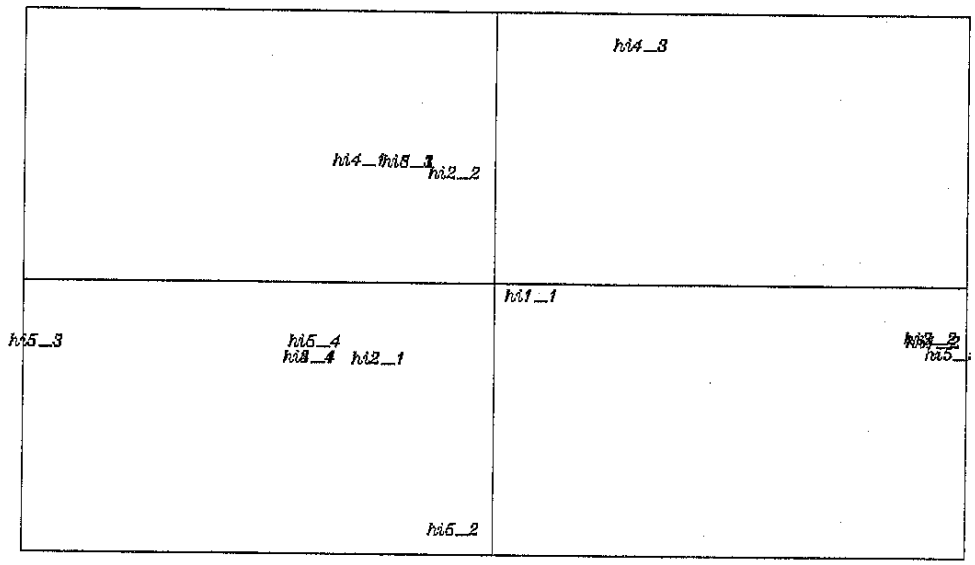

Figure 3.4: Modèles de 1 à 5 régimes — Haemophilius influenzae

Sur la figure 3.4, la situation est la même jusqu'à quatre régimes, mais le passage de quatre à cinq est moins évident ; il semble qu'au moins deux points du modèle à quatre régimes soient décomposés puis remplacés par trois nouveaux points. Les processus de bifurcation correspondant aux autres bactéries sont tous similaires à l'un ou l'autre de ces deux graphes : on pourra se reporter à l'annexe pour les étudier plus en détails.

Ce mécanisme de RHOM est intéressant ; il n'interviendra pas directement dans le reste de notre rapport puisque la suite de notre étude concerne des modèles à trois régimes nous y reviendrons à la fin de ce mémoire lorsque nous évoquerons des développements possibles de notre travail.

Puisque le découpage biologique que nous avons analysé précédemment comportait trois régimes, et que notre objectif est ici d'identifier la nature biologique des régimes obtenus par R'HOM, nous allons nous attacher maintenant au découpage statistique d'un modèle à trois régimes. On peut considérer d'abord le nuage formé par les trois points associés à chacune des onze bactéries étudiées :

A la lumière de l'étude du chapitre 2, on constate d'abord une symétrie assez nette par rapport à l'axe 1, ce qui est très encourageant. En outre, pour chaque espèce, on observe un point proche de l'axe 1, et deux points de part et d'autre de cet axe, tous deux d'abscisses proches de celle du premier point. On retrouve donc une disposition similaire à celle des régimes biologiques.

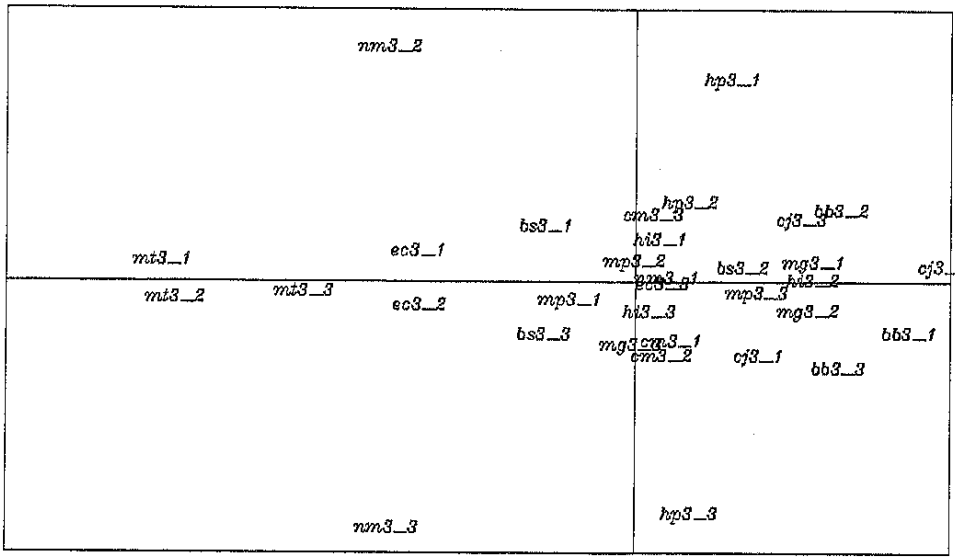

Figure 3.5: Modèle à trois régimes — Groupement par espèce

3.2.2 Groupement par régime biologique

En centrant les données, comme nous l'avons fait au chapitre 2, on obtient le graphe 3.6.

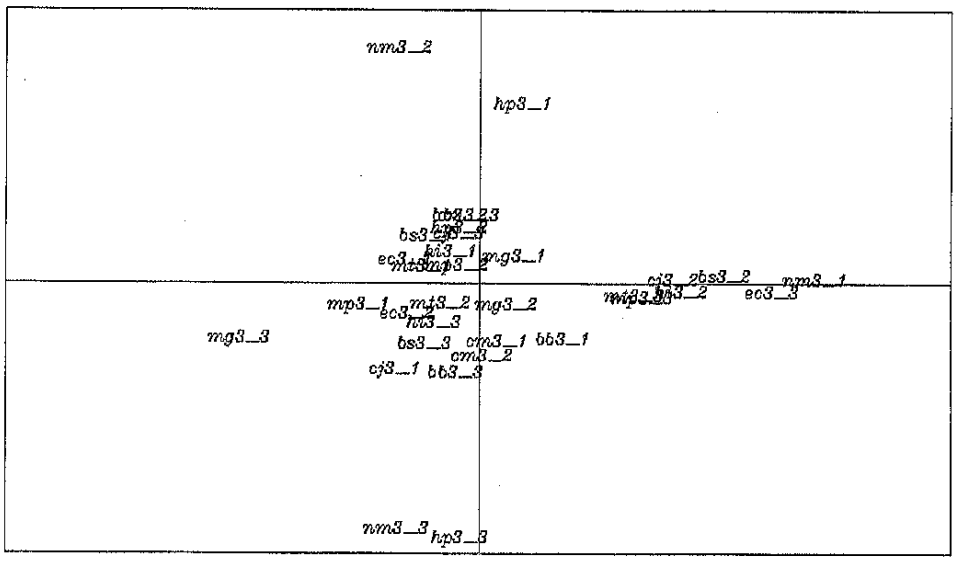

Figure 3.6: Modèle à trois régimes — Groupement par régime biologique

Bien que ce soit moins net qu'avec les régimes biologiques, on observe encore une fois un découpage en trois nuages relativement distincts :

- un premier nuage proche de l'axe, à droite de l'axe 2.
- un second, situé sous l'axe 1, et à gauche de l'axe 2.
- un troisième ensemble de points se distingue au dessus de l'axe 1, à gauche de

l'axe 2.

Concernant le regroupement par famille d'espèces, on n'observe pas non plus ici de nette tendance. On remarque une proximité remarquable entre certains points (mp3_1 et mg3_1 à gauche du graphe ; hi3_2 est le point le plus proche de ec3_3), mais rien qui permette d'identifier des points qui ne seraient pas étiquetés.

Chapitre 4

Rapprochement des deux analyses

4.1 Étiquetage biologique des régimes statistiques

4.1.1 Mise en place d'un étiquetage

Afin d'aller plus loin dans notre analyse, nous avons identifié biologiquement les régimes statistiques obtenus par R'HOM. Le principe de notre étiquetage est le suivant : pour chaque modèle estimé, R'HOM fournit, outre les matrices de transition associées aux régimes cachés, une sortie donnant, pour chaque position de la séquence, une estimation de la probabilité que la chaîne cachée soit dans chacun des régimes. Puisque les bases de données GenBank nous permettent de connaître le *régime biologique* réel de chaque base, on peut comparer cet étiquetage biologique à l'étiquetage statistique fourni par R'HOM.

Nous avons donc écrit un script BioPerl permettant d'extraire des fichiers GenBank le régime biologique de chaque base des génomes de nos 11 bactéries. Parallèlement, nous avons écrit un script Perl calculant, à partir des sorties de R'HOM, le régime statistique *le plus probable* associé à chaque base. Nous avons ensuite comparé les sorties à l'aide de SAS, en croisant les effectifs associés aux régimes biologiques et les effectifs associés aux régimes statistiques obtenus par R'HOM.

4.1.2 Résultats

Pour sept des onze bactéries étudiées, une correspondance très nette peut se faire entre les régimes biologiques et statistiques, ce qui nous a permis d'identifier les régimes statistiques. On pourra se reporter à l'annexe D pour le détail des tableaux croisés. Notons seulement que pour mycoplasma genitalium , le régime non-codant n'est pas identifiable directement ; en revanche, les deux autres régimes sont très bien identifiés, nous avons donc qualifié de "non-codant" celui des trois régimes statistiques qui n'était pas étiqueté.

En appliquant la procédure MDS aux sept espèces retenues, nous avons obtenu après centrage des données le graphe de la figure 4.1.

On remarque que les régimes se distinguent nettement, et qu'ils sont regroupés comme on pouvait l'imaginer :

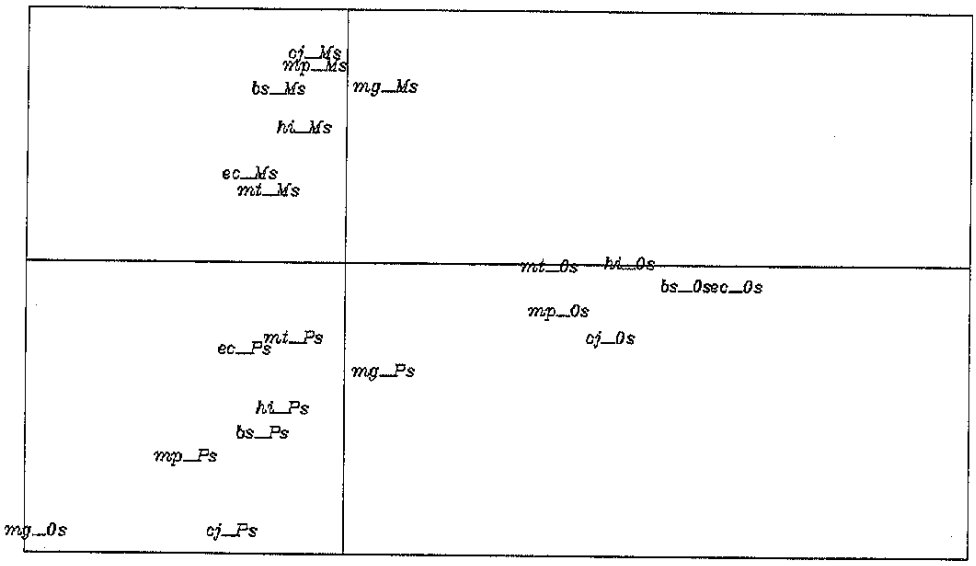

Figure 4.1: Modèle à trois régimes — Groupement par régime biologique (données étiquetées)

- le régime non-codant est situé à droite de l'axe 2 et légèrement sous l'axe 1.
- le régime codant négatif comprend des points situés sous l'axe 1, presque tous à gauche de l'axe 2.
- le régime codant positif est au dessus de l'axe 1, essentiellement à gauche de l'axe 2.

Remarquons enfin la position surprenante du point mg_0s ; elle s'explique par le fait que nous ayons retenu la bactérie mycoplasma genitalium (mg) bien que son régime non-codant soit très mal identifié.

Ces résultats sont très encourageants : ils suggèrent que si l'on dispose d'une séquence inconnue, on peut la découper en trois régimes par R'HOM, et identifier avec une probabilité significative la nature biologique des régimes.

4.2 Application à la prévision de la nature biologique d'un régime statistique

4.2.1 Objectif

Imaginons la situation dans laquelle un biologiste disposerait d'un génome non annoté, ou partiellement annoté. La démarche classique consiste à continuer l'annotation à l'aide de méthodes purement biologique.

Nos résultats permettent de penser que le biologiste pourrait utiliser le logiciel R'HOM pour identifier des plages statistiquement homogènes, afin d'établir une approximation des zones de recherches des gènes et des autres segments de code remarquables. Cette démarche engrangerait des gains de temps et de moyens sensibles, particulièrement importants vu la complexité du séquençage d'un génome.

4.2.2 Méthode envisagée

Techniquement, considérons un génome au mieux partiellement annoté. On peut lui appliquer la méthode des HMM avec R'HOM et obtenir un découpage en trois régime statistiques, auxquels sont associées les matrices de transition et les matrices stationnaires correspondantes. On peut alors utiliser ces nouveaux points en tant que variables supplémentaires dans la représentation par la procédure MDS fournie ci-dessus. On sera alors en état d'identifier ou non la nature la plus probable de chaque régime en fonction de la position de son point représentatif.

Pour obtenir des résultats fiables, il serait bon de pouvoir visualiser très clairement les nuages de points associés à chaque régime. Pour cela, il faudrait disposer d'un nombre plus importants de points, c'est-à-dire de plus de bactéries. Ceci est un prolongement possible de notre travail.

Une autre idée à retenir est que si le biologiste dispose d'une annotation partielle d'un génome, nos méthodes pourraient sans doute être ajustées pour utiliser au mieux les informations contenues dans les annotations. Réciproquement, puisque chaque modèle statistique conduirait à de nouvelles annotation, on pourrait imaginer une application dynamique des chaînes de Markov, en répétant la même démarche à plusieurs reprises, en tenant compte à chaque fois de l'état des annotations.

Bien sûr, on peut tempérer ce discours en arguant que dans notre cas, le modèle ne marche pas pour quatre des onze séquences, ce qui est beaucoup. C'est effectivement une marge d'erreur importante, mais si l'on réfléchit au problème du biologiste, la situation d'erreur revient simplement pour lui à étudier les séquences comme il le ferait sans l'aide de R'HOM. En d'autres termes, le travail statistique ne peu en aucun cas nuire à l'effort biologique. On peut donc penser que dans plus de la moitié des cas, le temps de travail nécessaire à l'identification des régions codantes est au moins divisé par deux, tandis que dans les autres cas, la démarche biologique demeure inchangée.

4.3 Comparaison directe des régimes biologiques et statistiques

Pour approfondir notre analyse, il serait intéressant maintenant de comparer directement la disposition des régimes statistiques et des régimes biologiques. Nous avons donc suivi la même procédure, mais en représentant six points par espèce : les trois points correspondants au découpage biologique utilisé au chapitre 2, et les trois régimes obtenus par R'HOM. On obtient alors le graphe suivant (L'étiquetage des régimes statistiques suit le même principe que pour les régimes biologiques, la notation xy_Rb est simplement remplacée par xy_Rb) :

On remarque tout d'abord que la modélisation statistique fournit des points beaucoup plus étalés : les points obtenus par l'analyse biologique des données sont beaucoup plus concentrés. Ce n'est pas étonnant, puisqu'ene modélisation par chaînes de Markov cachées repère des régimes ayant une forte homogénéité interne, et qui sont par conséquent très différenciés.

De manière assez surprenante, on constate ensuite une symétrie entre points biologiques et statistiques. Tous les points associés au régime non-codant sont toujours

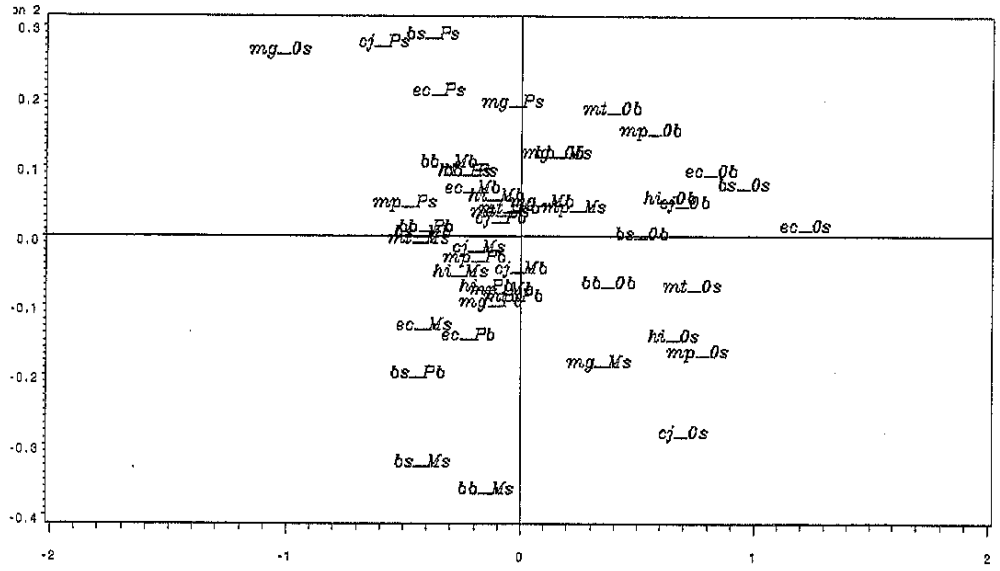

Figure 4.2: Modèle à trois régimes — Régimes biologique et statistiques

sur la partie droite du plan, mais beaucoup plus étalés suivant l'axe 2. Le régime codant positif ne se distingue pas nettement du codant négatif. On peut dire néanmoins que pour les points issus du modèle statistique, le codant positif est essentiellement situé à gauche de l'axe 2 et au dessus de l'axe 1, tandis que les points issus de l'analyse biologique sont concentrés dans le quadrant en bas à gauche, proches de l'origine. En ce qui concerne le régime codant négatif, la disposition est symétrique par rapport à l'axe 1.

Ainsi, pour une espèce donnée, un point *statistique* quelconque est plus proches des deux autres points statistiques (respectivement biologique), alors qu'on pouvait s'attendre à ce que pour chacun des trois régimes, les deux points associés soient les plus proches. Cela semble assez contradictoire, mais il ne faut pas négliger le fait que les graphes obtenus par la procédure MDS sont des projections. Il est donc possible que d'autres effets non mentionnés ici puisse primer sur la proximité des régimes. C'est pourquoi nous avons étudié les graphes associés à chacune des sept espèces retenues, dont voici un exemple. Les autres graphiques sont donnés en annexe ??.

Cette figure est caractéristique de l'ensemble des graphes obtenus ; les distances séparant les points semblent assez incohérentes. Or, cete situation est confirmée par l'étude directe des matrices des distances entre régimes.

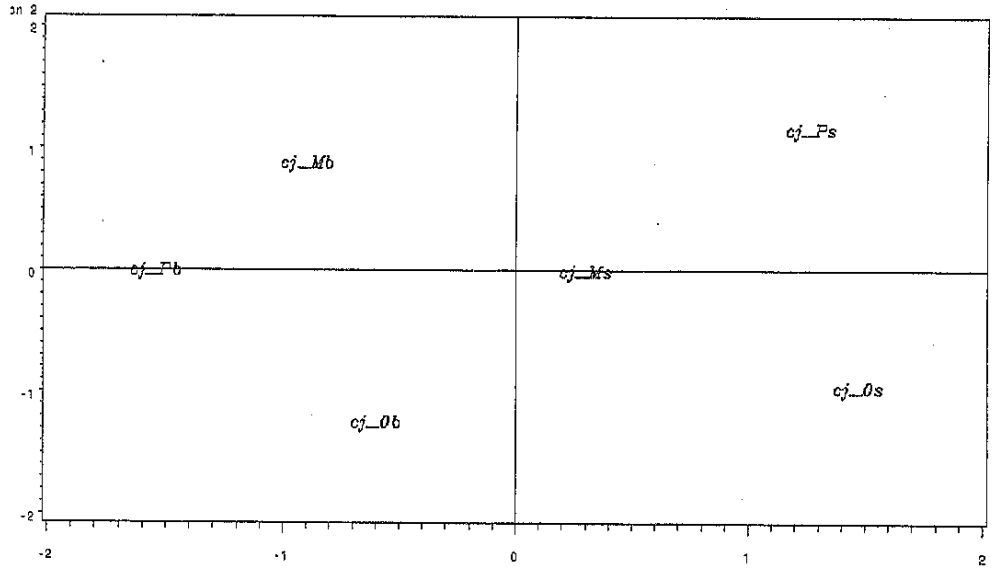

Figure 4.3: Régimes biologique et statistiques — Campylobacter jejuni

Conclusion

Ces derniers résultats sont surprenants, mais ils doivent être replacés dans le contexte de notre travail. D'une part, ils permettent de s'interroger sur le domaine de validité de notre démarche, sans pour autant remettre en question l'essentiel de nos résultats.

D'autre part, ils doivent être replacés dans le contexte mathématique de notre approche. En effet, les graphes que nous avons commentés sont obtenus par la procédure MDS, qui établit les distances entre régimes à partir de leurs matrices de transition et de leurs distributions stationnaires. Or les distributions stationnaires sont calculées par diagonalisation des matrices de transition, qui dépendent elles-mêmes du niveau de convergence de l'algorithme EM utilisé par R'HOM.

Les points sont donc construits par un processus dont chaque étape mériterait d'être étudiée théoriquement et empiriquement (à l'aide de méthodes de simulation par exemple). En particulier il serait intéressant d'étudier la sensibilité des lois stationnaires à de petites variations de la chaîne sur laquelle l'estimation est faite. En effet, une légère modification d'une fréquence de transition très petite pourrait entraîner des variations importantes de la loi stationnaire. Ceci permettrait peut-être d'expliquer pourquoi deux régimes markoviens estimés sur des zones qui coïncident à 95% peuvent avoir des lois très différentes, et donc être très distants en variation totale.

Par ailleurs, nous nous sommes essentiellement intéressés aux matrices de transitions fournies par R'HOM. Une étude statistique des autres sorties du logiciel, à savoir les positions des régimes, peut être envisagée, et a d'ailleurs été menée dans [1].

Annexes

Annexe A

Extraction des données

A.1 Exemple de fichier GenBank (extrait)

A.2 Complétion des bases de données GenBank

```
#include <stdio.h> #include <string.h> #include <stdlib.h> #include <time.h>      // pour initialiser la fonction rand


char choix(double  p, char a, char b);

int main(int argc,char *argv[]) {

        FILE *infile;
        FILE *outfile;
        FILE *logfile;

        const int buff  = 200;
        char* str       = malloc(buff*sizeof(char));                  // contenu de la ligne courante
        char chr_rpl;                                                 // caractère par lequel on remplace
        char* str_cm    = ">";                                        // délimiteur de commentaires

        char* log       = malloc(buff*sizeof(char));                  // nom du fichier de log
        char* str_log   = malloc(buff*sizeof(char));                  // contenu de la ligne courante
        char* str_tmp   = malloc(sizeof(char));

        double a, t, g, c;                                            // nombre de bases de chaque type
        int i;
        int rpl=0;

        const int taille= 7;
        const int prof  = 3;
        char** tab;

        double * proba  = malloc(taille*sizeof(double));              // proba de la premiere possibilite

        srand( (unsigned) time( NULL ) );

        a = atof(*(argv+argc-4));
        c = atof(*(argv+argc-3));
        g = atof(*(argv+argc-2));
        t = atof(*(argv+argc-1));

// Initialisation des tableaux

        tab = malloc(taille*sizeof(char*));             // caractères a remplacer
        for(i=0; i<taille; i++) tab[i] = (char*)malloc(prof*sizeof(char));

        tab[0][0]='r'; tab[0][1]='a'; tab[0][2]='g'; proba[0]=a/(a+g);
```

```
        tab[1][0]='y'; tab[1][1]='c'; tab[1][2]='t'; proba[1]=c/(c+t);
        tab[2][0]='k'; tab[2][1]='g'; tab[2][2]='t'; proba[2]=g/(g+t);
        tab[3][0]='m'; tab[3][1]='a'; tab[3][2]='c'; proba[3]=a/(a+c);
        tab[4][0]='s'; tab[4][1]='g'; tab[4][2]='c'; proba[4]=g/(g+c);
        tab[5][0]='w'; tab[5][1]='a'; tab[5][2]='t'; proba[5]=a/(a+t);
        tab[6][0]='n'; tab[6][1]='r'; tab[6][2]='y'; proba[6]=(a+g)/(a+g+t+c);

        switch( argc )                                          // nombre d'arguments de la ligne de commande
        {
    case 7:                                                     // deux noms de fichiers, cinq entiers
                if( (infile=fopen(argv[1],"rb")) == NULL )      // fichier d'entree non accessible en lecture
                {
                        perror("Impossible de lire le fichier d'entree");
                exit(1);
                }
                if( (outfile=fopen(argv[2],"wb")) == NULL )     // fichier de sortie non accessible en ecriture
                {
                        perror("Impossible d'ecrire dans le fichier de sortie");
                        exit(1);
                }
                strcpy (log,argv[2]);
                strcat (log,".log");
                if( (logfile=fopen(log,"wb")) == NULL )         // fichier de log non accessible en ecriture

                {
                        perror("Impossible d'ecrire dans le fichier de log");
                        exit(1);
                }

                break;
    default :
                printf("\nSyntaxe : complete.exe <fichier d'entree> <fichier de sortie> <a> <c> <g> <t>\n");
                return(0);
                exit(1);
        }
// lecture du fichier ligne par ligne tant qu'il n'est pas fini
        while (fgets( str , buff, infile ) != NULL)
        {
// si ce n'est ni une ligne de commentaire ni une ligne vide
                if (    ( strncmp(str,str_cm,strlen(str_cm))!=0 )
                        && ( strncmp(str," ",strlen(" "))   !=0 ))
                {
// tant qu'elle ne contient pas que des atgc
                        while ( strspn( str, "atgc" ) < strlen( str )-1 )
                        {
// définition du caractère de remplacement
                                i=0;
                                while ((i<taille)&&(tab[i][0]!=*(str+strspn( str, "atgc" )))) {i++;}
                                if (i==taille)          // le caractère n'était pas parmi ceux repertories
                                {
                                        printf("Attention, il y avait un %c\n",*(str+strspn( str, "atgc" )));
                                        chr_rpl = 'n';
                                }
                                else
                                        chr_rpl = choix(proba[i],tab[i][1],tab[i][2]);
// on remplit la log
                                *str_tmp = *(str+strspn( str, "atgc" ));
                                strcpy (str_log,str_tmp);
                                strcat (str_log," -> ");
                                *str_tmp = chr_rpl;
                                strcat (str_log,str_tmp);
                                if (*(str+strspn( str, "atgc" )) != 'n' )
                                {
                                        strcat (str_log,"\n");
                                        rpl++;
                                }
```

```
                                else strcat(str_log,", puis ");
                                fputs(str_log,logfile);
// remplacement
                                *(str+strspn( str, "atgc" )) = chr_rpl;
                        }
                }
// la ligne est desormais correcte, on peut la copier
                fputs(str,outfile);
        }
// fermeture du fichier d'entree
        fclose(infile);
// fermeture du fichier de sortie
        fclose(outfile);
// fermeture de la log
        fclose(logfile);
// liberation de la memoire
        for(i=0;i<taille;i++) free(tab[i]);
        free(tab);
        free(proba);
        free(str);

        printf("%d caractères remplacés\n",rpl);
        return(1);
}
```

Annexe B

Analyse des données biologiques

Les graphiques qui suivent détaillent la position respectives des régimes des bactéries pour le codant positif, le codant négatif, et le non-codant.

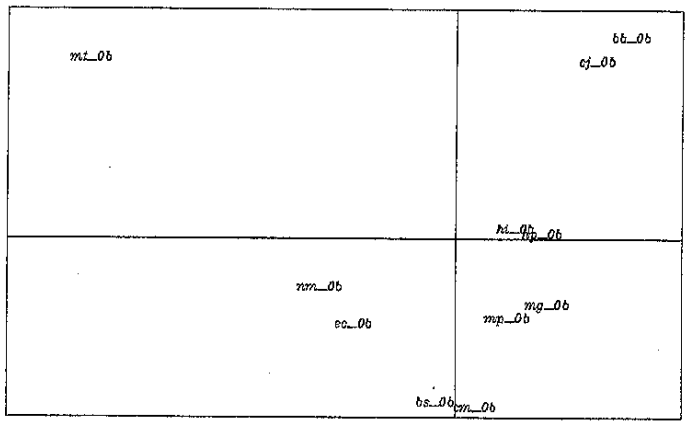

Figure B.1: Non codant

Les régimes du codant négatif ont la même disposition spatiale que ceux du codant positif.

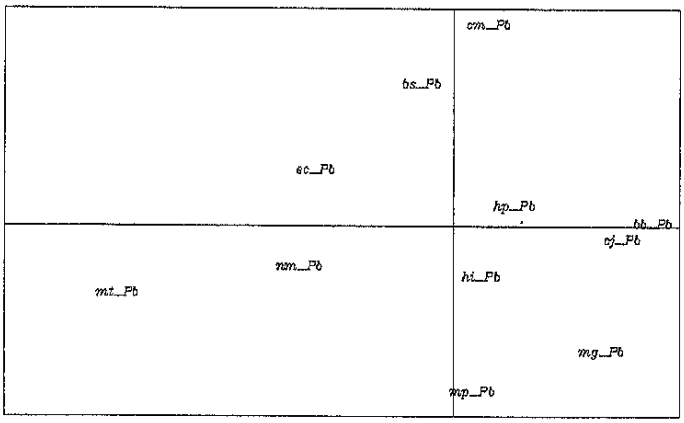

Figure B.2: Codant positif

Annexe C

Analyse des données statistiques

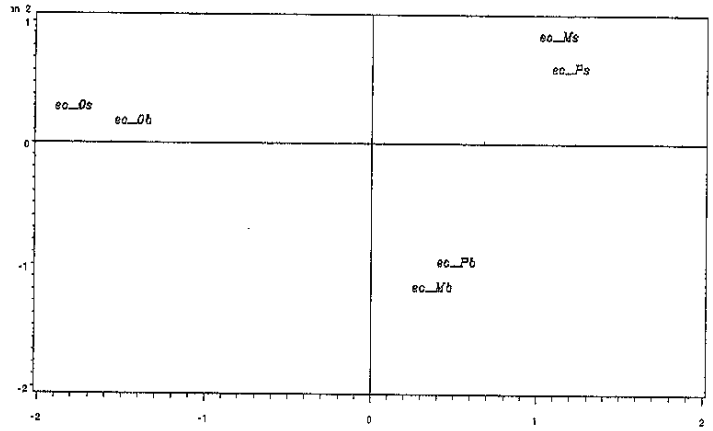

Figure C.1: Régimes biologiques et statistiques — Escherichia Coli

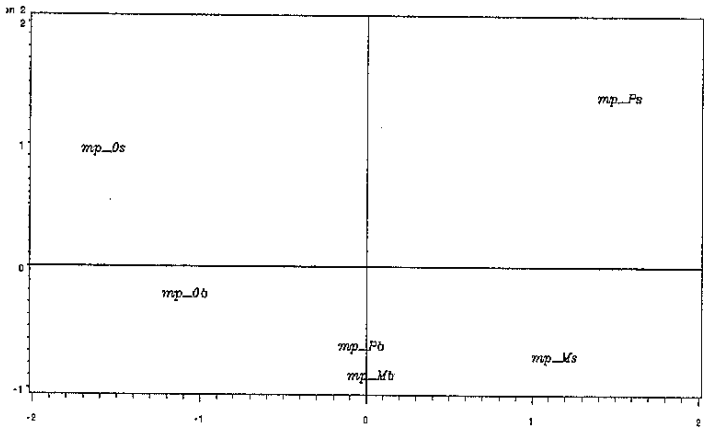

Figure C.2: Régimes biologiques et statistiques — Mycoplasma pneumoniae

Annexe D

Étiquetage biologique des régimes statistiques

hinfluenzae

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	757411	41.3855	95.3505	90.8973
1	Non codant	30856	1.6860	3.8845	16.2561
1	Codant +	6077	0.3321	0.7650	0.7530
2	Codant -	68637	3.7504	27.3508	8.2372
2	Non codant	128629	7.0284	51.2566	67.7665
2	Codant +	53685	2.9334	21.3926	6.6519
3	Codant -	7212	0.3941	0.9189	0.8655
3	Non codant	30327	1.6571	3.8641	15.9774
3	Codant +	747304	40.8332	95.2170	92.5952

mtuberculosis

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	1527154	34.6778	91.9880	73.1142
1	Non codant	60533	1.3746	3.6462	18.7376
1	Codant +	72479	1.6458	4.3658	3.6384
2	Codant -	79986	1.8163	4.4944	3.8294
2	Non codant	73061	1.6590	4.1053	22.6156
2	Codant +	1626643	36.9370	91.4004	81.6565
3	Codant -	481585	10.9356	49.9580	23.0564
3	Non codant	189462	4.3022	19.6541	58.6468
3	Codant +	292933	6.6518	30.3879	14.7051

mpneumoniae

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	5270	0.6455	1.9475	1.7501
1	Non codant	11884	1.4557	4.3916	12.5849
1	Codant +	253451	31.0452	93.6609	60.2249
2	Codant -	261677	32.0528	84.7191	86.9007
2	Non codant	18750	2.2967	6.0704	19.8558
2	Codant +	28449	3.4847	9.2105	6.7600
3	Codant -	34175	4.1861	14.4251	11.3492
3	Non codant	63797	7.8145	26.9285	67.5594
3	Codant +	138941	17.0189	58.6464	33.0151

mgenitalium

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	209344	36.0892	83.5468	88.4932
1	Non codant	16430	2.8324	6.5570	35.4469
1	Codant +	24797	4.2748	9.8962	8.3447
2	Codant -	23549	4.0597	8.8477	9.9546
2	Non codant	12840	2.2135	4.8242	27.7017
2	Codant +	229771	39.6106	86.3281	77.3228
3	Codant -	3672	0.6330	5.7970	1.5522
3	Non codant	17081	2.9446	26.9659	36.8514
3	Codant +	42590	7.3422	67.2371	14.3324

bsubtilis

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	1611598	38.2365	92.9537	84.5522
1	Non codant	80456	1.9089	4.6405	18.1311
1	Codant +	41711	0.9896	2.4058	2.2365
2	Codant -	255196	6.0547	35.2685	13.3888
2	Non codant	277505	6.5840	38.3517	62.5370
2	Codant +	190879	4.5288	26.3798	10.2346
3	Codant -	39244	0.9311	2.2330	2.0589
3	Non codant	85784	2.0353	4.8811	19.3318
3	Codant +	1632441	38.7310	92.8859	87.5289

ecoli

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	2052813	37.1318	93.3387	85.0428
1	Non codant	106182	1.9206	4.8280	15.9239
1	Codant +	40322	0.7294	1.8334	1.6473
2	Codant -	30166	0.5457	1.3374	1.2497
2	Non codant	120786	2.1848	5.3552	18.1141
2	Codant +	2104540	38.0675	93.3074	85.9776
3	Codant -	330881	5.9851	30.8187	13.7075
3	Non codant	439840	7.9559	40.9673	65.9620
3	Codant +	302915	5.4792	28.2139	12.3751

cjejuni

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	4870	0.2967	0.8772	0.6716
1	Non codant	20223	1.2320	3.6428	24.8354
1	Codant +	530056	32.2913	95.4800	63.4877
2	Codant -	30032	1.8296	10.0123	4.1414
2	Non codant	47638	2.9021	15.8819	58.5032
2	Codant +	222282	13.5416	74.1059	26.6239
3	Codant -	690256	42.0508	87.7764	95.1870
3	Non codant	13567	0.8265	1.7252	16.6613
3	Codant +	82557	5.0294	10.4984	9.8883

hpylori

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	16658	0.9988	2.7280	2.0849
1	Non codant	38378	2.3010	6.2849	27.4918
1	Codant +	555605	33.3123	90.9872	76.1843
2	Codant -	414533	24.8541	75.1356	51.8829
2	Non codant	50550	3.0308	9.1624	36.2111
2	Codant +	86630	5.1941	15.7020	11.8787
3	Codant -	367787	22.0513	72.7552	46.0322
3	Non codant	50670	3.0380	10.0235	36.2971
3	Codant +	87056	5.2196	17.2213	11.9371

cmuridarum

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	192398	17.9317	73.3708	41.0128
1	Non codant	31463	2.9324	11.9984	34.7903
1	Codant +	38366	3.5757	14.6308	7.4730
2	Codant -	5748	0.5357	1.8316	1.2253
2	Non codant	15839	1.4762	5.0470	17.5140
2	Codant +	292242	27.2372	93.1214	56.9232
3	Codant -	270971	25.2548	54.5330	57.7619
3	Non codant	43134	4.0201	8.6807	47.6956
3	Codant +	182789	17.0361	36.7863	35.6038

bburgdorferi

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	138300	15.1857	76.7272	32.6061
1	Non codant	16494	1.8111	9.1507	32.1064
1	Codant +	25455	2.7950	14.1221	5.8491
2	Codant -	280325	30.7805	61.6469	66.0904
2	Non codant	22082	2.4247	4.8561	42.9837
2	Codant +	152320	16.7252	33.4970	35.0002
3	Codant -	5529	0.6071	2.0051	1.3035
3	Non codant	12797	1.4051	4.6408	24.9100
3	Codant +	257422	28.2656	93.3541	59.1507

nmeningitidis

Régime statistique	Régime biologique	Effectif	Fréquence	Fréquence colonne	Fréquence ligne
1	Codant -	107060	4.9011	25.1023	10.8145
1	Non codant	216518	9.9120	50.7670	57.6861
1	Codant +	102916	4.7114	24.1307	12.5645
2	Codant -	411882	18.8556	46.8123	41.6057
2	Non codant	79974	3.6611	9.0894	21.3072
2	Codant +	388002	17.7624	44.0983	47.3691
3	Codant -	471023	21.5630	53.6440	47.5798
3	Non codant	78846	3.6095	8.9796	21.0067
3	Codant +	328185	15.0240	37.3764	40.0664

Bibliographie

[1] Guillaume Barruel, Pierre-Yves Bourguignon, Romuald Elie, and Jérémie Jakubowicz. Mémoire de statistique appliquée. 2001.

[2] Jeff A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. 1998.

[3] G. Churchill. Stochastic models for heterogeneous dna sequences. *Bulletin of Mathematical Biology*, 1989.

[4] Ian de Leeuw and Willem Heiser. Theory of multidimensional scaling. *Handbook of Statistics*, 2, 1982.

[5] A. Dempster. Maximum likelihood from incomplete date via the em algorithm. *Journal of the Royal Statistical Society*, 1977.

[6] Vincent Miele. 13 bioperl rares. 2002.

[7] Pierre Nicolas, Philippe Bessieres, Anne-Sophie Tocquet, Kevin Bryson, Bernard Prum, Florence Muri, and Franáois Rodolphe. Apprentissage automatique de modèles de chaînes de markov cachées pour la détection des gènes bactériens.

[8] Pierre Nicolas and Florence Muri-Majoube. R'hom, recherche d'homogénéités dans une séquence d'adn. 2001.