UNIVERSITÉ EVRY VAL D'ESSONNE
ECOLE DOCTORALE "DES GÉNOMES AUX ORGANISMES"

THÈSE

en vue de l'obtention du grade de
Docteur de l'université d'Evry Val d'Essonne
dans la spécialité Mathématiques appliquées
préparée au sein du Laboratoire Statistique et Génome

et présentée par
PIERRE-YVES BOURGUIGNON

PARCIMONIE DANS LES MODÈLES MARKOVIENS
ET APPLICATION À L'ANALYSE DES SÉQUENCES BIOLOGIQUES

sous la direction du Pr. BERNARD PRUM

Soutenue le 15 décembre 2008, devant le jury composé de :

JEAN-PHILIPPE VERT, président
GILLES CELEUX, rapporteur
ANDRÉ BERCHTOLD, rapporteur
IVO GROSSE, examinateur
BERNARD PRUM, directeur de thèse

Table des matières

Préface

Cette thèse, ainsi que le mémoire de DEA qu'elle prolonge, ont été le moyen d'une transition radicale dans ma formation scientifique : initialement formé comme ingénieur statisticien économiste, mon immersion dans les problématiques issues de la biologie s'est progressivement effectuée par le biais des méthodes statistiques pour la génomique, et plus généralement de ce grand ensemble hétérogène de méthodes que l'on appelle la *bioinformatique*.

Aborder la biologie par ce biais fut délicat. Un génome est probablement la donnée expérimentale la plus éloignée qui soit de la biologie d'un organisme, tout en conditionnant cette dernière d'une manière considérée comme ultime. Ce paradoxe fait de la génomique un point d'entrée peu efficace vers la biologie pour le novice, mais il est contrebalancé par l'extraordinaire diversité des questions auxquelles la bioinformatique tente de répondre : les congrès de bioinformatique offrent à ce titre une diversité de thématiques biologiques rarement égalées dans les conférences de biologie, typiquement plus spécialisées, et constituent une source de curiosité et d'émerveillement presque inépuisables.

Cette diversité peut effrayer au premier abord. Comment en effet acquérir la moindre pertinence dans un domaine que plusieurs années d'études académiques ne parviennent pas à couvrir intégralement lorsque l'on y entre totalement novice ? Comment identifier, dans cette masse d'information, ce qui est essentiel de ce qui est anodin ? L'analyse des génomes cristallise ce questionnement. Les génomes constituent en effet la donnée expérimentale la plus fiable de la biologie, contiennent a priori la totalité de l'information nécessaire à ce que la vie ait lieu... sauf qu'il faut savoir identifier cette information parmi les millions de nucléotides que représentent déjà un génome bactérien typique.

Les génomes portent de multiples niveaux d'informations : certes ils portent les modèles des protéines, codés au travers du code génétique dans les gènes, mais ils portent aussi les modèles de multiples ARN non-codants, ainsi qu'une diversité quasiment inépuisable de motifs qui conditionnent leurs interactions (et celles de leurs transcrits) avec les autres molécules. Et tout cela en se conformant aux contraintes métaboliques de production des nucléotides nécessaires à la réplication du génome, ou encore des acides aminés utilisés dans les protéines. Grâce à une certaine lenteur de l'évolution, ces signaux sont plus ou moins partagés entre les organismes vivants, et ont pu être identifiés par recoupement avec des résultats expérimentaux acquis depuis les années 1950, date des débuts de la biologie moléculaire. En termes statistiques, cela signifie que l'on procède de manière supervisée : on recherche ce qui est commun à un ensemble de séquences, tout en les distinguant d'un autre ensemble. Cependant, la conservation évolutive n'est jamais exacte, et l'on comprend que rapidement que s'il est une règle en biologie, c'est celle de la diversité. Là commence la nécessité du recours aux statistiques.

Dans le cas particulier de l'analyse des génomes, les approches statistiques consistent principalement à caractériser un ensemble de séquences (ou de régions d'un génome) par sa composition en nucléotides, voire en k-mots de nucléotides, éventuellement en tenant compte de la position des nucléotides dans les séquences. Et lorsque l'on commence à explorer la littérature en la matière, on réalise rapidement que les chaînes de

Markov constituent un modèle de référence en la matière. Ce sont en effet a) des modèles sur l'espace des séquences, b) dont la statistique exhaustive minimale est le vecteur des comptages des mots d'une longueur donnée. Ils admettent en fait une justification bien plus ultime, formulée dans la première partie de ce manuscrit, une *bonne* raison pour laquelle les chaînes de Markov forment bien *le* modèle à utiliser pour modéliser la composition en mots de séquences, qui résulte de l'application du principe de maximum d'entropie au cadre particulier de l'espace d'états des séquences, muni d'observables que sont les nombres d'occurrences de chacun des mots d'une longueur donnée.

Mais le choix d'une distribution statistique qui capture les régularités recherchées repose fondamentalement sur le choix des observables, autrement dit de la longueur des mots dont l'on dénombre les occurrences. Et, dans le cas particulier des séquences biologiques, on ne dispose pas toujours d'un a priori sur la longueur des mots dont la distribution est biaisée dans l'ensemble de séquences d'intérêt. L'enjeu réside donc dans la problématique du choix de l'ordre des chaînes de Markov, et renvoie à une problématique de sélection de modèles.

La sélection de l'ordre d'une chaîne de Markov doit résoudre un compromis. Augmenter l'ordre du modèle revient en effet à diviser l'échantillon que constitue la séquence (finie) en autant de sous-échantillons que de distributions conditionnelles à estimer, et de ce fait détériore la qualité de chacun de leurs estimateurs. En revanche, cela permet de capturer des corrélations de plus longue portée dans la séquence. La détérioration de la qualité des estimateurs se traduit par le risque de sur-apprentissage, puisque le maximum de vraisemblance atteignable pour un échantillon augmente avec l'ordre du modèle, quitte à estimer certaines distributions conditionnelles avec un échantillon très réduit. Le modèle rend alors potentiellement compte d'artefact de l'échantillon. Diverses approches permettent de résoudre ce compromis, par exemple testant la significativité de l'accroissement de la vraisemblance de l'échantillon par l'ajout d'un paramètre, ou encore en pénalisant la vraisemblance d'une quantité croissante avec la dimension du modèle.

On peut cependant s'interroger sur ce qui constitue un *bon* compromis. La consistance de ce compromis, c'est-à-dire sa capacité à sélectionner correctement un modèle ayant généré un échantillon avec une probabilité tendant vers 1 avec la taille de l'échantillon, est un critère important. Mais sous cet argument, de nombreux critères de sélection de modèles permettent un *bon* compromis. Qui plus est, il s'agit d'un argument asymptotique. Il s'avère que la statistique bayésienne fournit un argument plus riche, qu'il est possible d'instancier sous la forme d'un critère à optimiser. Nous présenterons dans la deuxième partie une construction de cet argument, dans le cadre des méthodes de compression de texte où il a été principalement développé pour le cas particulier des séquences, et présenterons également les techniques d'optimisation de ce critère introduites pour la compression.

Cependant, le choix d'un critère de sélection de modèles n'est pas le seul ingrédient de la modélisation des séquences. Le choix de la classe de modèles au sein de laquelle un modèle en adéquation avec les données est recherché est également un élément essentiel. Un grand pas a été franchi dans la direction de l'enrichissement de la classe des chaînes de Markov lors de l'introduction de chaînes de Markov à longueur variables, ou arbres de contexte, au début des années 1980. En définissant un ensemble de contextes structuré en arbre, plutôt qu'une longueur fixe pour l'ensemble de ceux-ci, il devenait possible de prendre en compte les dépendances d'une manière adaptative, permettant ainsi une modélisation raisonnable des motifs peu fréquents dans les séquences. Comme nous le verrons dans la dernière partie de ce manuscrit, il est possible d'enrichir encore la classe de modèles des arbres de contexte en permettant la fusion de nœuds

dans l'arbre. Cette extension, que J. RISSANEN, le pionnier des arbres de contexte, semblait souhaiter ajouter dans son article fondateur, n'avait à ce jour fait l'objet d'aucune investigation. Aussi, la dernière partie de ce manuscrit constitue une contribution originale. Plus encore que la définition de cette classe de modèles étendue, nous proposons une extension de l'algorithme bayésien *Context Tree Maximization* à ce nouveau cadre, et dérivons un algorithme de sélection de modèle permettant la maximisation exacte de la probabilité a posteriori du modèle.

Ce cheminement intellectuel n'a pas été aussi linéaire que ce manuscrit le présente. Cherchant dans un premier temps à adapter les méthodes d'estimation des chaînes de Markov cachées de manière à insérer une sélection de modèles des lois d'émission comme chaînes de Markov à longueur variable, certaines faiblesses de celles-ci sont apparues, et le recours à la maximisation d'une vraisemblance pénalisée (approche privilégiée par BUHLMANN pour réaliser la sélection de ces modèles) s'est révélé un obstacle de taille pour l'intégration de ces approches dans les algorithmes EM. Sous la houlette de P. NICOLAS, nous nous sommes alors tournés vers la problématique de la fusion des nœuds. La granularité plus fine des modèles résultant, ainsi que l'absence de certains des écueils des VLMC de ce cadre plus général, nous ont convaincu de son intérêt. La contrepartie étant, bien entendu, un coût algorithmique plus important, mais également l'éloignement de la perspective d'intégrer ces approches dans les modèles à variable cachée (puisqu'il nous fallait, dans un premier temps, construire un cadre théorique autour de cette idée de fusion des nœuds dans les arbres de contexte). A cette époque, la divergence des terminologies entre statistique classique et théorie de l'information aidant, nous n'avions pas identifié les liens étroits qu'il existait entre les chaînes de Markov à longueur variable et arbres de contexte. Aussi passions-nous à côté des puissants algorithmes issus de la compression, CTM et CTW. Et c'est avec une approche bayésienne que nous pensions nouvelle pour ces problématiques que nous avons construit un critère de sélection de modèle, pour lequel j'ai ensuite obtenu un résultat de convergence.

Ce n'est qu'à l'issue de ce travail (et des trois années de financement de cette thèse) que j'ai rencontré P. COLLET, physicien statisticien à l'Ecole Polytechnique, et que je me suis familiarisé avec les travaux de E. T. JAYNES qui m'avaient déjà été mentionnés par un bioinformaticien issu de la physique statistique, N. LARTILLOT. Ce moment fut la source d'une profonde satisfaction intellectuelle, ce dernier présentant en effet une approche constructive des modèles statistiques, à partir de fondements profonds et dont certaines ramifications m'échappent encore. Il m'est apparu particulièrement satisfaisant de procéder à la construction des chaînes de Markov en suivant cette démarche, tâche qui, bien que mentionnée à diverses reprises dans la littérature, n'est que peu explicitée. Disposant de ce cadre conceptuel qui fournissait enfin une justification non-asymptotique à la modélisation statistique, l'exploration de ses liens avec la statistique bayésienne, dont les fondements m'apparaissent aujourd'hui comme le stade ultime du raisonnement jaynésien, et dont j'étais un utilisateur ignorant tel M. JOURDAIN de la prose, a accompli cette (re)construction à partir des fondements des méthodes statistiques que j'avais mises en œuvre auparavant.

Cependant, ce cheminement n'est pas encore accompli : il me reste le regret (que je souhaite dépasser dans les prochaines années) de ne pas avoir su dériver les résultats de convergence des estimateurs à partir des résultats de concentration de l'entropie. Cette notion m'échappe encore quelque peu à ce jour, ce qui explique son absence de l'exposé qui suit. Un autre aspect plus fondamental encore qui m'apparaît comme une lacune à ce jour est la compréhension de la géométrie sous-jacente à la maximisation de l'entropie, ainsi qu'aux approches bayésiennes. La maximisation de l'entropie sous les contraintes des observations recourt en effet aux méthodes lagrangiennes d'optimisation, dont les coordonnées associées correspondent précisément à ce que

l'on nomme le *paramètre* du modèle en statistique classique. Cette remarque permet a priori de transposer les résultats de la géométrie lagrangienne à ce cas particulier d'une variété structurée par l'entropie, et je reste curieux de voir les résultats qui pourraient être obtenus en poursuivant cette approche. Qui plus est, l'extension au cadre bayésien (autrement dit, la prise en compte de la taille finie des échantillons) de cette approche géométrique pourrait s'avérer utile, en particulier pour les questions de sélection de modèle. Celles-ci mettent en effet en jeu invariablement l'*expansion de* LAPLACE, qui repose elle-même fondamentalement sur des résultats de convexité.

Ces années passées à l'appropriation des fondements des approches statistiques, mais aussi à l'apprentissage de l'informatique, de l'enseignement supérieur, et surtout de la biologie, n'auraient pas porté les mêmes fruits sans le cadre chaleureux et vivant du laboratoire Statistique et Génome. A ce titre, comme à des milliers d'autres que je ne listerai pas ici, je suis tout particulièrement reconnaissant à B. PRUM d'avoir patiemment permis, accompagné, et soutenu cette démarche. Je remercie également V. MIELE et M. HOEBEKE pour leur intervention salvatrice à tous les niveaux de la programmation des algorithmes présentés ici ; C. MATHIAS, dont l'imperturbable exigence scientifique m'a beaucoup apporté ; H. RICHARD pour son enthousiasme jamais démenti au cours de nos trois années de présence commune au sein de ce laboratoire, et surtout P. NICOLAS, sans qui ce travail n'aurait probablement jamais vu le jour. Je tiens enfin à remercier tout particulièrement V. SCHÄCHTER, responsable du groupe *Réseaux métaboliques* du Génoscope, ainsi que *J. Weissenbach*, directeur de cet institut, qui m'y accueillent depuis trois ans dans d'excellentes conditions, et m'ont permis de procéder à la réécriture de ce manuscrit tout en achevant ma conversion à la biologie.

Première partie

L’analyse statistique des séquences biologiques

Introduction

Depuis l'identification de l'acide désoxyribonucléique comme support de l'information génétique dans les années 1950, suivie du développement de diverses technologies permettant, entre autres, la détermination de sa séquence ou la modification contrôlée de celle-ci, l'interprétation de cette information est un enjeu d'une importance croissante. Aujourd'hui, les technologies de séquençage les plus récentes permettent en effet la détermination de plusieurs milliards de bases en une journée en utilisant un seul appareil...

Ainsi, plusieurs centaines de génomes microbiens complets sont aujourd'hui déposés dans les bases de données publiques, et l'on s'attend à atteindre le millier dans les prochains mois (voir [**?**]). Malgré cela, la proportion de gènes jusque là inconnus identifiés dans les nouveaux génomes séquencés stagne aux alentours de 50% : ce chiffre témoigne du retard grandissant de l'acquisition d'informations expérimentales permettant l'interprétation des génomes séquencés.

Pour y remédier, de nombreuses attentes sont posées sur les méthodes bioinformatiques. Au-delà du fil de développement principal qu'a connu cette jeune discipline dans les deux dernières décennies, à savoir la détection des régions codantes et des signaux les flanquant dans les séquences génomiques, l'enjeu aujourd'hui se situe en aval : il s'agit à présent d'exploiter au mieux la connaissance acquise sur les gènes connus pour la généraliser de manière optimale aux nouvelles séquences. De l'efficacité de ces méthodes dépend en effet l'efficacité de l'allocation des ressources expérimentales consacrées à l'identification de nouvelles fonctions protéiques, et en particulier enzymatiques. Le projet présenté par le *fellowship for the interpretation of genomes* ([**?**]) résume bien cet aspect de l'évolution de la discipline.

Cette première partie revient sur les origines des méthodes de détection de gènes, puis présente une justification du recours au modèle probablement le plus utilisé dans la modélisation de séquences biologiques aujourd'hui, les chaînes de MARKOV (éventuellement cachées). Enfin, muni de cet outil statistique, le dernier chapitre présente la manière dont il est mis en œuvre pour diverses applications à l'étude des séquences des macromolécules biologiques.

CHAPITRE 1

Les éléments de la vie

1. Les cellules

1.1. Des systèmes hors de l'équilibre. Le vivant est constitué de matière, mais se distingue pourtant de la matière inerte par le fait qu'il présente une structure, qui a la particularité de se maintenir de manière auto-catalytique, et même de se dupliquer de manière tout aussi auto-catalytique. L'élément primordial de cette structure propre au vivant est la cellule, constituant élémentaire de l'intégralité du vivant, des bactéries aux eucaryotes supérieurs[1].

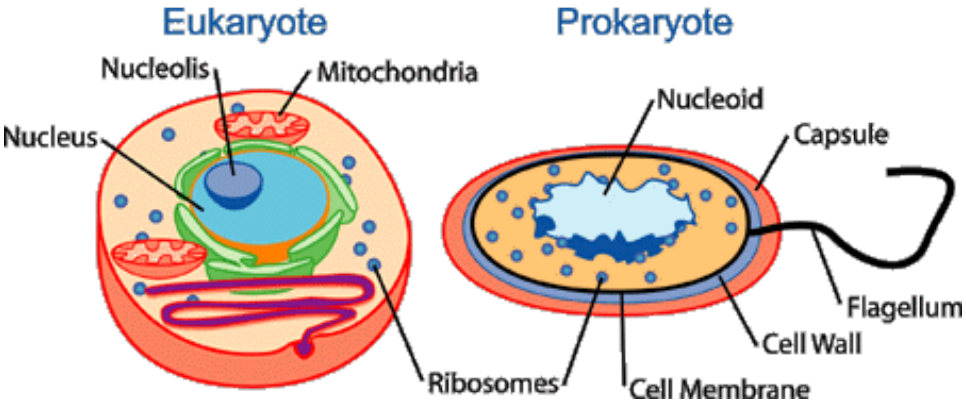


FIG. 1. Schémas de la structure d'une cellule eucaryote (à gauche) et procaryote (à droite).

Une cellule est avant tout constituée d'une paroi (voir figure 1), qui délimite un milieu intérieur et un milieu extérieur. En séparant ces deux milieux, la paroi permet la capture de matière dans le milieu intérieur, et par suite de maintenir des gradients avec l'extérieur. Il peut s'agir de gradients de concentrations d'entités chimiques, mais aussi de gradients de quantités physiques telles que la pression. La cellule est un système physico-chimique qui se maintient hors de l'équilibre avec son environnement.

Comme tout système hors de l'équilibre vis-à-vis de son environnement, la cellule est le siège de multiples changements d'état : conversions chimiques, déformations, sont autant de conséquences spontanées de l'absence d'équilibre. D'après la thermodynamique, chacun de ces changements d'état tend à ramener la cellule vers l'équilibre avec son environnement. Le miracle du vivant, s'il en est un, est sa capacité à ralentir ce retour à l'équilibre : la cellule, aussi longtemps qu'elle vit, parvient à maintenir sa structure, et, partant, son déséquilibre avec l'environnement. La reproduction permet par ailleurs une forme de juvénation, le *désordre* accumulé au long de la vie n'étant

[1]Cette définition exclut les virus, mais ces derniers n'ont pas de capacité de reproduction auto-catalytique propre, puisque c'est le fonctionnement de leur hôte qui les reproduit. Cependant, l'ensemble des organismes vivants exploitent une forme de symbiose avec leur environnement, puisque les écosystèmes dans lesquels ils peuvent survivre sont ceux qui maintiennent l'environnement assurant le départ de l'équilibre physico-chimique.

pas transmis à la descendance. La nature parvient à ce résultat en canalisant les réactions spontanées qui ont lieu en son sein, de telle manière que leur avancement maintient le déséquilibre avec l'environnement avec une efficacité suffisante pour laisser le temps à la reproduction d'avoir lieu. L'ensemble de ces réactions de conversion chimique du milieu afin d'en extraire l'énergie et l'information nécessaire à la vie est désigné par le terme de *métabolisme*. On peut aujourd'hui avoir une vision globale, ou intégrée, de la masse de connaissances accumulée par les biochimistes sur le métabolisme grâce aux réseaux métaboliques qu'ils ont reconstitué pour certains organismes. A titre d'exemple, la figure 2 représente une portion du réseau métabolique de la bactérie *Escherichia coli*, en l'occurence la voie des pentose phosphates.

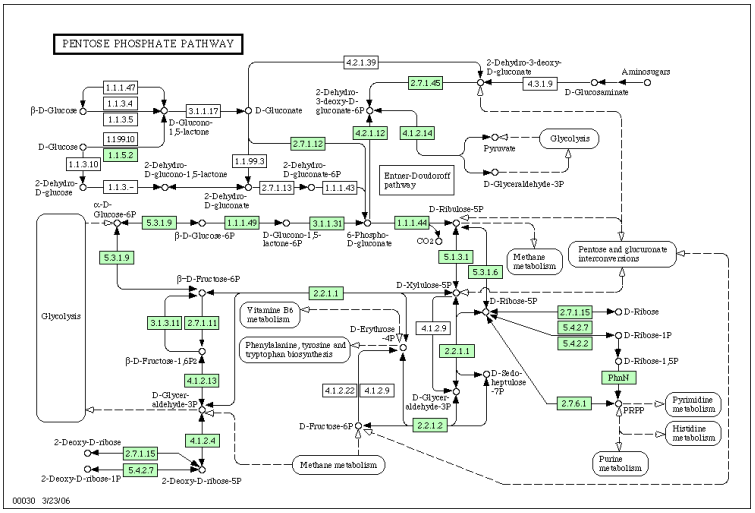


FIG. 2. Exemple de réseau métabolique : la voie des pentose phosphate de la bactérie *Eschrichia Coli. Source : Kyoto Encyclopedia of Genes and Genomes (KEGG)*, `http ://www.genome.ad.jp/kegg`

Il est bien entendu que l'environnement se trouve modifié par le fonctionnement métabolique de la cellule, et qu'en cas de ressources limitées, les cellules finiront par ne plus trouver y trouver les ressources nécessaires au maintien de ce déséquilibre. C'est ce qui arrive dans une boîte de Petri dans laquelle on laisse se développer une colonie bactérienne : lorsque la totalité des nutriments ont été consommés par les cellules pour croître et se reproduire, il ne reste à leur disposition que les résidus de la dégradation de ces nutriments. Elle entrent alors dans une phase de croissance ralentie, dénommée *phase stationnaire* par opposition à la *phase exponentielle* caractéristique de l'excès de nutriments. Leur métabolisme fonctionne alors avec un rendement plus faible, et consomme typiquement les résidus produits par la croissance exponentielle pour en tirer l'énergie qu'ils contiennent encore.

1.2. Les effecteurs de la cellule. Au delà de la possession d'une paroi, la cellule doit donc être capable de contrôler les échanges à travers cette paroi, ainsi que les conversions chimiques qui ont lieu dans son milieu intérieur. Car si toutes les réactions pouvaient avoir lieu de la même manière que dans le milieu extérieur, elle ne saurait les exploiter de manière à maintenir sa structure.

Du point de vue thermodynamique et informationnel, cela implique pour la cellule d'être capable d'exploiter le paysage de potentiels physico-chimiques que lui impose l'environnement. Elle réalise cette tâche au moyen d'*effecteurs*, le plus souvent des molécules très particulières et propres au vivant, qui ont la capacité de catalyser plus ou

moins spécifiquement diverses réactions, et qui ainsi façonnent le paysage des potentiels physico-chimiques à l'intérieur de la cellule. Par le choix de ces effecteurs, la cellule opère une sélection des réactions qui composent son métabolisme. D'ailleurs, extrêmement peu de réactions qui ont lieu dans une cellule sont spontanées, au sens où elles ne peuvent avoir lieu à un rythme comparable en l'absence des enzymes qui les catalysent. Il s'agit d'un exemple frappant de la nécessité d'une mémoire biologique, la structure d'un réseau métabolique résultant de millions d'années d'élaboration sous l'action de l'évolution.

L'action des effecteurs est donc orientée vers un double objectif : le maintien de la cellule hors de l'équilibre physico-chimique avec son environnement, ce qui a pour effet de permettre la poursuite de la maintenance et de l'expression de l'information génétique.

1.3. Métabolisme. La conversion physico-chimique de l'environnement met en jeu une quantité importante de molécules spécialisées, qui réalisent chacune une étape élémentaire de ce processus de conversion. Ces molécules sont pour la quasi-totalité des *protéines* : ce sont de très grosses molécules (on parle de *macromolécules*, qui sont capables de catalyser de manière plus ou moins spécifique une réaction physico-chimique donnée, et qui sont synthétisées par la cellule au cours de sa vie.

Dans une bactérie, on trouve dores et déjà plusieurs milliers de protéines différentes. Cependant, seule une fraction contribue effectivement à la conversion du milieu environnant : on les appelle les *enzymes*. Les autres protéines réalisent d'autres tâches sur lesquelles nous reviendront. Chacune de ces enzymes catalyse une étape bien précise de la conversion de l'environnement, depuis la dégradation des *métabolites*[2] pour en extraire de l'énergie et des composés élémentaires, jusqu'à la synthèse des éléments constitutifs de la cellule, tels que les *acides aminés*, précurseurs de la synthèse des protéines, ou encore l'assemblage des lipides pour former la paroi de la cellule.

Chaque enzyme intervient donc à des instants bien précis de la conversion : il faut en effet que d'autres enzymes aient déjà permis la production de ses substrats, et que d'autres encore consomment ensuite les produits de la réaction qu'elle catalyse, comme le montre le réseau métabolique de la figure 2. Les différentes réactions s'enchaînent donc pour former un réseau de réactions chimiques, qui, ensemble, permettent la vie. Ce *réseau métabolique* diffère d'une espèce à une autre, mais des similarités importantes sont partagées par les réseaux de la plupart des espèces. Ainsi, on retrouve des pièces de ce réseau[3], que l'on appelle des *voies métaboliques*, dans les réseaux d'organismes très divers : cellules humaines, bactériennes et de plantes possèdent de nombreuses voies métaboliques communes, qui sont les traces physiologiques de leur évolution divergente depuis un ou plusieurs ancêtres communs. A ce titre, le projet Reactome (voir `HTTP://WWW.REACTOME.ORG`, et [**?, ?, ?, ?**]) fournit incidemment une illustration éloquente de ce phénomène : parmi les 2530 réactions identifiées chez l'homme (incluant des réactions non métaboliques, par exemple liées à la signalisation ou la régulation), organisées en 818 voies, 316 sont retrouvées[4] chez la bactérie la plus étudiée, *Escherichia coli*, recouvrant environ 200 voies.

Par ailleurs, tout organisme doit être prêt à vivre dans des environnements différents. Si les plantes, par exemple, sont adaptées au sol, à la température, etc. qu'elles trouvent dans l'écosystème où elles poussent, elles le sont aussi aux variations régulières de cet environnement induites par les saisons, ou par tout autre cause. Il en de

[2]Ce terme désigne les petites molécules importées (ou exportées) par la cellule de (vers) son environnement, par opposition aux grosses molécules synthétisées par le vivant.

[3]moyennant quelques variations, *cf* la notion de variant métabolique introduite dans [**?**]

[4]Reactome est une initiative centrée sur l'annotation des processus biologiques chez l'humain, mais une projection systématique des annotations sur d'autres organismes, par homologie de séquence, est effectuée pour chaque nouvelle version de la base de données.

même des bactéries, qui sont capables de persister dans des environnements très divers pour certaines. Bien entendu, selon l'environnement rencontré, une cellule ne peut pas utiliser les mêmes portions de son réseau métabolique : si un composé extrêmement pratique pour produire des acides aminés est régulièrement absent de son environnement de croissance, il faut bien qu'elle possède un autre moyen de produire ces acides. Elle exploite alors une voie métabolique alternative, qui aura probablement un rendement plus faible.

La vision du métabolisme telle que les réseaux la proposent est donc éminemment statique. Elle ne rend pas compte de sa *plasticité*, c'est-à-dire de la capacité de la cellule à activer ou inactiver certaines réactions selon l'environnement, son état, et éventuellement l'état de ses congénaires dans la colonie. Cette plasticité est réalisée par des mécanismes complémentaires du métabolisme, qui permettent l'évaluation de l'état de l'environnement (la *signalisation*) et une réponse adéquate en terme de production des protéines nécessaires à la survie dans cet environnement (la *régulation*). Les effecteurs de ces mécanismes sont aussi des protéines ou des molécules d'ARN, pour lesquels des modèles sont également stockés sur l'ADN.

2. Les protéines

Nous avons évoqué précédemment le rôle des enzymes, dont la fonction tient aux conversions chimiques de petites molécules qu'elle catalysent, mais celles-ci ne représentent qu'une portion de l'ensemble des protéines présentes au sein d'une cellule. Que font les autres ? Tout le reste, ou presque, pour que les enzymes fonctionnent correctement. Ce qui demande de synthétiser les enzymes, et plus précisément de synthétiser les bonnes enzymes au bon moment.

2.1. Les différentes fonctions. Au-delà des fonctions enzymatiques, les protéines doivent donc remplir une foule d'autres tâches qui permettent à l'activité métabolique de se dérouler. Nous n'évoquerons ici que les principales.

2.1.1. *Signalisation.* Tout d'abord, lorsqu'une espèce chimique que la cellule peut métaboliser se présente dans l'environnement, il faut que la cellule l'identifie. Ce travail de signalisation est effectuée par des protéines dont la conformation change en présence du métabolite. A la suite de ce changement de conformation, l'affinité de la protéine vis-à-vis d'autres protéines ou d'autres métabolites est modifiée de telle manière que le signal peut être transmis à d'autres protéines. Typiquement, la forme active de la protéine se comporte comme un *facteur de transcription*, activant l'expression d'un certain nombre de gènes *cibles*, et, partant, la production par la cellule des protéines nécessaires à la métabolisation de ce composé. En pratique, une protéine de signalisation peut activer la transcription de plusieurs autre protéines, et en particulier d'autres facteurs de transcription. On parle alors de *cascade régulatoire*. Par ailleurs, les protéines qui ont une activité de facteur de transcription peuvent également *réprimer* l'expression de certains gènes, par exemple de manière à inhiber une voie métabolique au profit d'une autre, plus rentable en présence du métabolite détecté.

Ces relations d'activation/répression de l'expression entre gènes font l'objet de nombreuses études aujourd'hui. Ces relations sont abstraites sous la forme de graphe appelés *réseaux de régulation*, dans lesquels une arête orientée relie deux gènes si celui placé à sa source influence l'expression de celui placé à sa cible. Les arêtes peuvent par ailleurs être différenciées, selon que l'influence qu'elles représentent est activatrice ou inhibitrice. Dans ce cadre formel, de nombreux travaux fournissent des outils permettant de caractériser la dynamique de tels réseaux, de manière qualitative comme quantitative.

Parallèlement, la reconstruction des réseaux de régulation d'organismes naturels se développe depuis quelques années sur la base de *données d'expression*, mesures de la quantité d'ARN messagers de chacun des gènes d'un organisme, et reste à ce jour un challenge. Plus récemment, des méthodes d'analyse des séquences promotrices des

gènes permettant l'identification d'un grand nombre de sites de fixation de facteurs de transcription ont été développées, et fournissent une information complémentaire aux données d'expression. Enfin, il existe des technologies de validation expérimentale de l'affinité d'un facteur de transcription avec des sites de fixation inférés, mais elles restent limitées en terme de débit.

2.1.2. *Synthèse de l'enzyme : transcription.* Les intermédiaires inévitables de ces cascades régulatoires sont les facteurs de transcription. Ceux-ci se lient à la molécule d'acide désoxyribonucléique (ADN), en des sites bien précis. Au voisinage de ces sites, la molécule d'ADN porte un modèle de l'enzyme à produire. En présence du facteur de transcription, un second type de protéine, l'ARN polymerase (voir figure 3, se lie à l'ADN et transcrit le *modèle* en une molécule formée d'acide ribonucléique (ARN). Cette opération est réellement une copie, au sens où elle est neutre vis-à-vis de la manière dont est codée l'information génétique : un ARN est également une séquence de radicaux pris parmi 4, tout comme l'ADN, et il existe une bijection entre la séquence des nucléotides portée par la molécule d'ADN et la séquence de la molécule d'ARN, qui n'est autre que la bijection qui assure la complémentarité des brins. La seule différence notoire est le remplacement des T de la molécule d'ADN en uracyle (U), forme stable lorsque la molécule n'est pas double brin, comme l'ARN l'est.

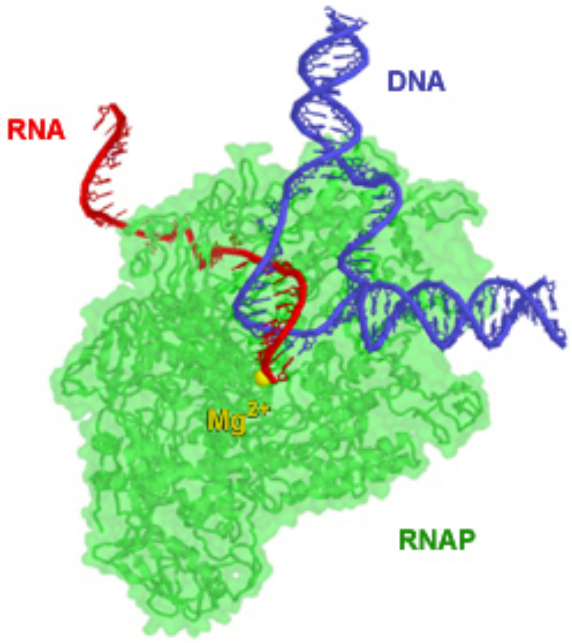


FIG. 3. Image d'un complexe composé d'une ARN Polymérase liée à un brin d'ADN et un brin d'ARN qu'elle produit conformément à la séquence d'ADN. L'ARN polymérase est un complexe protéique composé le plus souvent de 5 sous-unités chez les bactéries.

2.1.3. *Synthèse de l'enzyme : traduction.* Cette molécule d'ARN se lie à son tour à un complexe protéique très massif, le *ribosome*, qui assemble la protéine à partir des acides aminés, et ce conformément au modèle recopié du brin d'ADN. Le ribosome se saisit de la molécule d'ARN en un site particulier, appelé *site de fixation du ribosome* (ou RBS pour *ribosome binding site* en anglais) comme il se doit (voir figure 4).

A ce stade, il est essentiel pour la protéine que l'ensemble des 20 acides aminés différents soient disponibles pour l'assemblage de la protéine. En effet, le défaut d'un acide

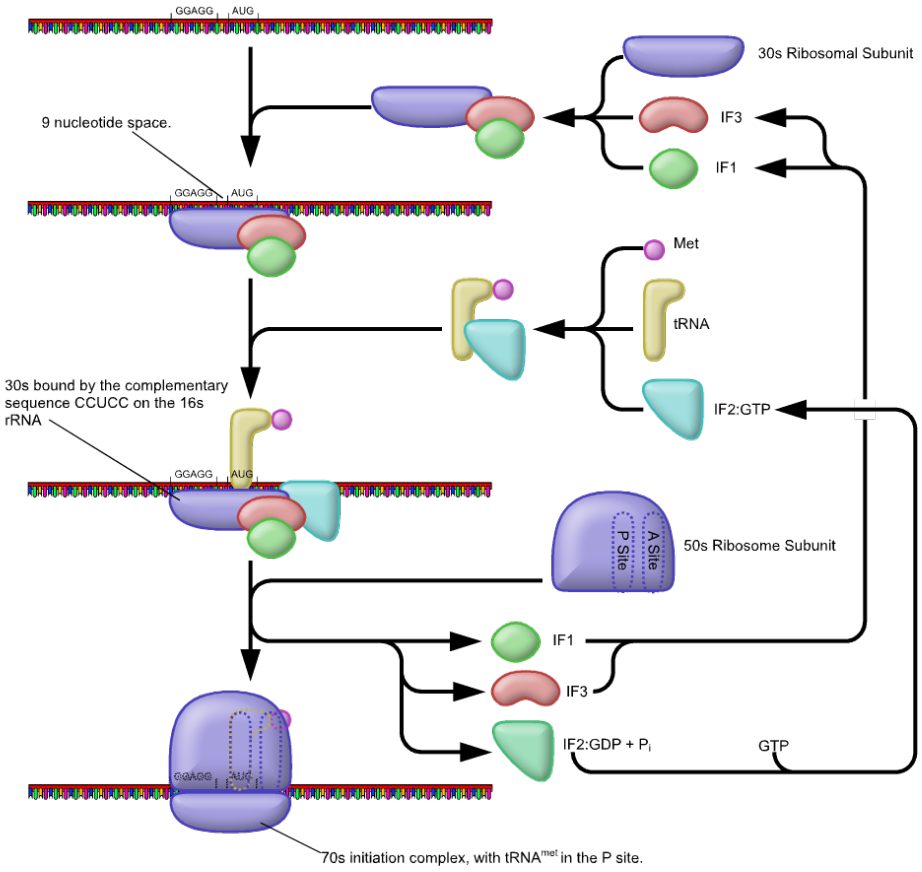


FIG. 4. Le processus d'initiation du la traduction de l'ARN en protéine : le ribosome de fixe sur le brin d'ARN en un site spécifique, puis initie la traduction de la séquence d'ARN en protéine.

aminé requis pour la synthèse conduirait inévitablement à la synthèse d'une protéine erronnée, qui risquerait de ne pas avoir l'activité enzymatique prévue et perturberait le fonctionnement de l'ensemble du réseau métabolique. Il a été ainsi montré que des organismes ayant évolué longtemps dans un environnement limité en soufre s'étaient adapté à ces conditions, en particulier en fixant des mutations dans un certain nombre de gènes de manière à éviter l'utilisation d'acide aminés soufrés pendant la synthèse des protéines correspondantes.

On distingue sur la figure 4 que la sous-unité 50S du ribosome comporte deux sites particuliers, désignés par les lettres A et P. Ce deux sites sont au contact de la séquence d'ARN, et *voient* chacun un triplet de nucléotides. Ces sites accueillent les ARN de tranfert (ou ARN$_t$) (voir figure 6), qui assurent l'assemblage de la protéine conformément au code génétique : en effet, ils portent à une extrémité une séquence de trois nucléotides que l'on appelle *anticodon*, et à l'autre extrémité un acide aminé spécifié par l'anticodon selon le code génétique (voir figure 5). L'adéquation entre l'anticodon et l'acide aminé dont un ARN de transfert est chargé est assurée par les ARN$_t$-synthétases, protéines spécialisés pour chaque acide aminé, et qui assurent la liaison de l'acide aminé

sur le site d'attachement des ARN de transfert correspondants. Certaines sont capables d'hydroliser la liaison lors d'une erreur d'appariement.

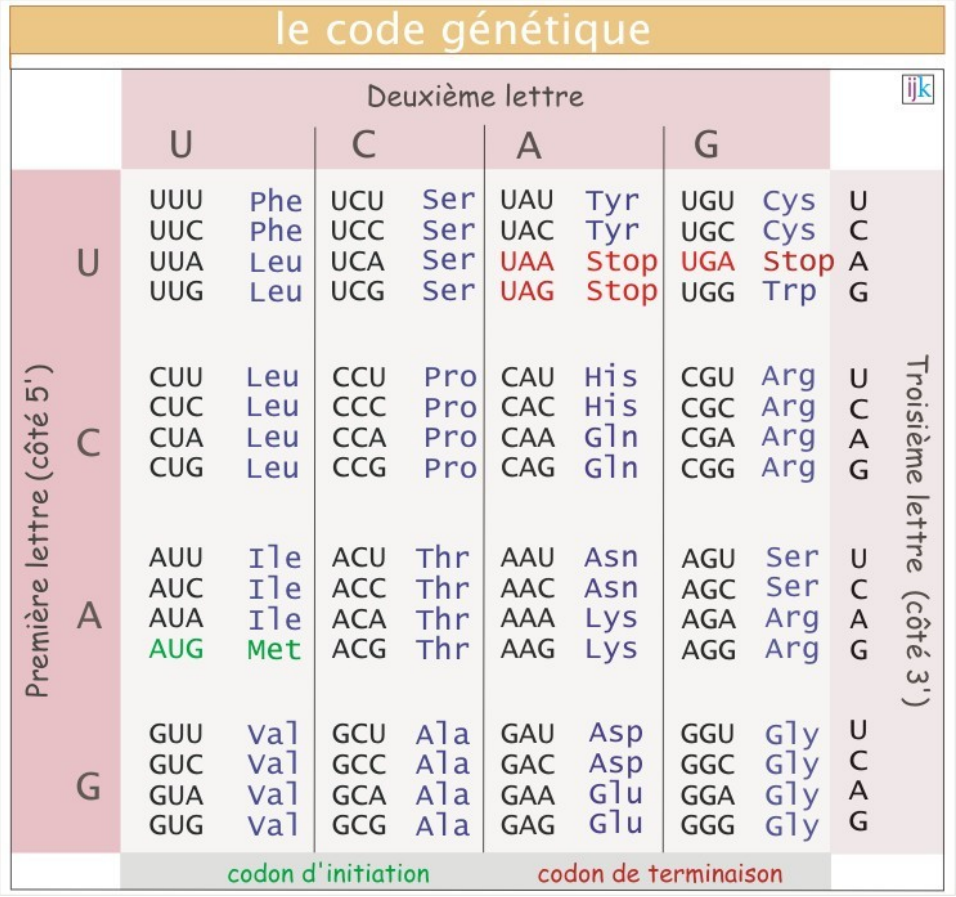


FIG. 5. Le code génétique spécifie la correspondance entre les codons portés par la molécule d'ARN et l'acide aminé à inclure pour former la protéine.

Le ribosome, lorsqu'il capture un ARN de transfert dans son site A, s'assure de la complémentarité : s'il est bien complémentaire du codon en regard sur le brin d'ARN, il se lie à celui-ci ; dans le cas contraire, il est rejeté par le ribosome. L'ARN de transfert fixé glisse alors vers le site P, où il est déchargé de son acide aminé, qui viendra allonger la chaîne de peptides déjà formée. Un dernier site, E, expulse l'ARN de transfert déchargé. Le ribosome assemble ainsi récursivement une protéine en fonction de sa séquence codée telle que recopiée depuis la molécule d'ADN, et selon le code génétique qui dicte la manière dont la séquence d'un gène doit être traduite en une séquence d'acides aminés.

2.1.4. *Synthèse de l'enzyme : repliement.* Une fois l'enzyme produite conformément au modèle décrit par l'ADN, il reste encore à lui faire adopter la conformation adéquate pour qu'elle soit fonctionnelle. Cette étape, dénommée *maturation* de la protéine, diffère entre les organismes : plutôt simple chez les procaryotes, elle est beaucoup plus complexe chez les eucaryotes, où le processus de maturation se déroule dans un compartiment particulier appelé *appareil de Golgi* (voir figure 1). A nouveau, d'autres protéines interviennent dans ce processus pour *chaperonner* le repliement de la protéine synthétisée.

L'organisation spatiale des sites catalytiques est en effet primordiale dans la fonctionnalité d'une protéine. Mal repliée, elle peut non seulement ne pas assurer sa fonction dans l'organisme, mais également interagir de manière parasite avec le métabolisme de l'organisme.

Par ailleurs, l'importance de l'organisation spatiale des sites catalytiques d'une protéine justifie de chercher à prédire cette structure à partir de la séquence elle-même, et de fournir ainsi une information de grande valeur pour prédire les fonctions des protéines dont le rôle est inconnu. Cet objectif est loin d'être atteint, et suscite une recherche foisonnante et en progrès constants.

3. L'acide désoxyribonucléique

3.1. De la caractérisation à l'analyse approfondie. A l'origine de la synthèse de toute protéine dans une cellule, nous avons vu qu'il y a un modèle de celle-ci porté par une molécule très particulière : l'acide désoxyribonucléique. Décrite pour la première fois au milieu du $19^{ème}$ siècle, cette molécule a été soumise à de nombreuses analyses visant à établir son rôle de support de l'information génétique. Au milieu du $20^{ème}$ siècle, un large faisceau d'indices permettait dores et déjà de lui attribuer ce rôle, mais sans que son fonctionnement soit décrit. Cette molécule fut l'objet de diverses analyses chimiques, qui permirent en particulier d'en déterminer la composition (voir [**?, ?**]). C'est alors que fut rapportée l'égalité des compositions en Adénine et Thymine d'une part, et en Guanine et Cystéine d'autre part.

C'est en 1953 que WATSON et CRICK ont validé un modèle de sa structure, modèle selon lequel la molécule est formée d'une double hélice dont l'adhésion des brins est assurée par la complémentarité des nucléotides : une molécule d'adénine se lie à une

FIG. 6. Une molécule d'ARN de transfert reproduite en trois dimensions : en haut à droite, l'*anticodon*, séquence de trois lettres que le ribosome place en regard du codon courant du brin d'ARN ; en bas à gauche, le site de fixation de l'acide aminé correpondant.

thymine sur le brin complémentaire, et une guanine à une cystéine. Cette complémentarité venait expliquer les mesures de la composition de l'ADN réalisées auparavant. Cependant, l'élucidation de la structure de l'ADN a mis en lumière sa propriété essentielle : son extraordinaire stabilité. Grâce à la redondance de l'information due à l'homologie des deux brins complémentaires, mais aussi grâce à sa structure de double-hélice, cette molécule est très résistante aux mutations. Mais au-delà d'un modèle de sa structure, les observations de WATSON et CRICK leur ont permis d'identifier le mécanisme d'expression de l'information génétique : l'idée qu'un gène est exprimé en ARN, puis traduit en protéine lorsqu'il s'agit d'un gène codant est née de leurs observations. Cette conception de la nature de l'information génétique introduisit une relation quasi bijective entre le génome et l'ensemble des protéines d'un organisme, relation que l'on appelle aujourd'hui le *dogme central de la biologie moléculaire*.

Dès lors, les découvertes se sont accumulées rapidement. En particulier, moins de vingt ans ont suffi à caractériser l'ensemble des protéines clefs dans les processus de réplication (découverte de l'ADN-polymérase) de l'ADN et d'expression des gènes (ARN-polymérase et ribosome, mais aussi les ARN de transfert qui convoient les acides aminés). La maîtrise d'une importante partie de cette machinerie cellulaire à donné lieu à l'ensemble des technologies actuelles mises en œuvre dans un laboratoire de génétique moléculaire :

- la *réaction de polymérisation en chaîne* (PCR pour *Polymerase Chain Reaction* en anglais), qui consiste à déclencher une succession de réactions de réplication de l'ADN, est une technique sans cesse raffinée pour amplifier des régions d'un génome, telle une photocopieuse dont le nombre de copies obtenues augmenterait exponentiellement avec le temps de fonctionnement. Elle permet d'obtenir une quantité de matériel génétique importante à partir de quelques (voir un seul) brins d'ADN, et en ce sens a permis le développement de technologies plus élaborées,
- le séquençage, qui utilise la même machinerie en la perturbant de manière à stopper la réplication au moins une fois en chacune des nucléotides, en insérant un marqueur fluorescent qui est incapable de lier une nucléotide supplémentaire. Il suffit alors de trier les séquences selon leur masse pour les ranger selon la longueur de la réplication qui a pu avoir lieu, et d'éclairer de manière à exciter les nucléotides marquées. En utilisant un marqueur différent pour chacune des nucléotides, on reconstitue ainsi la succession des nucléotides le long de la séquence.
- la transformation génétique utilise des enzymes capables de couper la molécule d'ADN. Ces enzymes sont à l'origine utilisées par les virus pour insérer leur séquence dans le génome de leur hôte, et ainsi bénéficier de son système de réplication pour se multiplier.

Muni de l'ensemble de ces méthodes, et de bien d'autres encore, la génétique moléculaire s'est attachée à comprendre la manière dont l'information biologique est encodée sur le génome. Or, si le code génétique, qui dicte la correspondance entre la séquence à 4 lettres d'un gène avec la séquence à 20 lettres de la protéine qu'il code, a été établi dès les années 1960, nombre de signaux portés par les génomes échappent encore à la sagacité des biologistes moléculaires. Aussi le chapitre suivant présente, de manière résumée et synthétique, l'ensemble des connaissances acquises sur la structure des séquences génomiques et protéiques, ainsi que les méthodes afférentes, qui constituent la boîte à outils de la *bioinformatique*.

3.2. La structure des génomes. Le long de la séquence d'un génome sont codées de multiples informations, au premier rang desquelles les gènes, c'est à dire les séquences des protéines que l'organisme est susceptible de produire. Face à une longue séquence issues de l'assemblage des lectures d'un séquençage, identifier les position

des gènes est la principale tâche requise pour l'exploitation des informations portées par le génome. Elle n'est pas trivial, car les régions d'un génome qui portent les gènes ne sont pas totalement *ponctuées* : si le codon STOP ne peut avoir pour effet que l'interruption de la traduction, il n'en va pas de même pour les codons START, qui, eux, peuvent également apparaître à l'intérieur d'une séquence codant une protéine. Le codon d'initiation peut en effet également être traduit en acide aminé (voir le code génétique, 5).

3.2.1. *Phases de lecture.* La caractéristique structurelle du code génétique la plus évidente est qu'il associe des triplets de nucléotides aux acides aminés qui composent la protéine. Compte-tenu du fait que le ribosome ne *lit* qu'une fois chaque nucléotide, cette structure implique qu'une séquence codante doit contenir un nombre de nucléotides multiple de 3. Cette propriété signifie aussi que qu'un même segment d'ADN peut coder plusieurs protéines. Il suffit en effet d'ôter la première nucléotide pour décaler d'une position la fenêtre de trois nucléotides lue par le ribosome. La séquence de codons lue par le ribosome est alors totalement différente. De même si l'on ôte encore une nucléotide au début de la séquence. En revanche, lorsque l'on ôte trois nucléotides, soit exactement la longueur d'un codon, on retrouve la même séquence de codons, sauf le premier bien sûr.

Il existe par conséquent trois manières de lire une région codante : en commençant en une position d'indice $3 \times k$, $3 \times k + 1$ ou $3 \times k + 2$. Et chacune de ces manières conduit à la traduction d'une protéine différente. On parle de *phase de lecture* ; un brin d'ADN peut être lu selon trois phases différentes. Ce raisonnement s'applique naturellement aux deux brins formant la molécule d'ADN, si bien qu'il existe six manières différentes de lire des séquences codantes dans une séquence d'ADN : trois manières pour chaque brin.

De même que les deux brins de la molécule d'ADN portent une information redondante (on peut déduire l'un de l'autre par les règles de complémentarité A–T et G–C), les séquences codantes lues sur deux phases différentes sont parfaitement redondantes. Cependant, les séquences de codons obtenues sur les six phases sont en général toutes différentes : des protéines différentes peuvent donc être codées simultanément (sur une même portion de la séquence) sur les six phases. D'autant que la dégénérescence du code génétique (le fait que plusieurs codons – en moyenne 3 – codent pour le même acide aminé) apporte une certaine flexibilité. Cependant, positionner deux séquences codantes sur la même région de l'ADN pourrait entraîner des collisions entre les machineries d'expression des gènes, et, par suite, une difficulté à produire certaines protéines. Seuls les virus, qui subissent une contrainte de compacité de leur génome, exploitent cette possibilité ; en revanche, bactéries, archaebactéries, et eukaryotes évitent cette situation. Tout au plus trouve-t-on quelques exemples de régions du génome où deux gènes sont codés sur les brins opposés.

3.2.2. *Cadres ouverts de lecture.* Face à un génome fraîchement séquencé, la première information recherchée est naturellement la position des gènes, et plus particulièrement des régions codantes. D'une certaine manière, il s'agit de retrouver la ponctuation du texte génomique, et heureusement quelques propriétés du code génétique (voir figure 5) sont d'un grand secours. Comme rappelé précédemment, une séquence codante doit commencer par le codon d'initiation de la traduction, sans lequel le ribosome ne commence pas la polymérisation de la protéine. Ce codon d'initiation de la traduction, que l'on appelle communément START, est le plus souvent codé par le codon AUG, mais il existe des exceptions relativement nombreuses à cette règle. Mais surtout, ce codon est également associé à un acide aminé via le code génétique : la présence du codon AUG à l'intérieur d'une région codante est donc possible. En revanche, le codon qui doit nécessairement clore la séquence codante, sans quoi le ribosome ne termine pas la traduction, que l'on nomme STOP, peut être codé de trois manière différentes : UAA, UAG, ou UGA, mais tous ces codons ne sont pas associés à des acides

aminés par le code génétique. Si bien que la présence d'un codon `STOP` en phase à l'intérieur d'une région codante n'est pas possible : ce codon terminerait la traduction. Ces codons sont ici formulés dans le contexte du brin d'ARN, aussi pour la séquence d'ADN ils sont respectivement `ATG` pour `START`, et `TAA`, `TAG` ou `TGA` pour `STOP`.

Grâce à cette propriété du codon `STOP`, il est possible de réaliser une approximation supérieure des régions codantes sur une phase de lecture du génome. Il suffit pour cela de détecter l'ensemble des codons `START` et `STOP` sur cette phase de lecture : les régions codantes sont nécessairement incluses dans les plus grands segments de longueur multiple de 3 commençant par un `START`, terminant par un `STOP` dans la même phase, et ne contenant aucun codon `STOP` en phase entre les deux. Un tel segment de la séquence d'ADN est appelée un *cadre ouvert de lecture (ou ORF, Open Reading Frame)*.

Les cadres ouverts de lecture forment une première liste de candidats pour les séquences codantes dans un génome. Cependant, cela vaut principalement pour les génomes de bactérie. En effet, les génomes eukaryotes peuvent déroger à cette règle à cause d'un phénomène appelé *décalage de la phase de lecture*. Ce phénomène est souvent associé à l'épissage d'une partie de l'ARN messager (partie que l'on appelle *intron*, et qui peut ne pas être systématiquement ôtée de la séquence de l'ARN messager) qui porte un nombre de nucléotides qui n'est pas multiple de 3.

Revenons donc aux génomes bactériens. Parmi cette liste de séquences codantes candidates que sont les cadres ouverts de lecture, certains sont de très petite taille : on considère qu'une protéine compte au minimum une cinquantaine d'acides aminés pour les plus petites, et une taille de 150 nucléotides est donc un minimum pour une séquence codante. Ce critère réduit encore la liste de candidats.

3.2.3. *Les signaux associés aux gènes.* Pour la réduire plus avant, il est nécessaire de prendre en compte la présence d'autres signaux. On distingue deux types de signaux : ceux portés par le cadre ouvert de lecture lui-même, et ceux portés par la *séquence flanquante* de ce cadre de lecture, et en particulier ceux portés par la séquence amont du cadre (dans le sens de lecture du brin).

Les signaux situés en amont de la séquence codante sont associés à la machinerie d'expression génétique de la cellule. En effet, l'expression de l'information portée par la séquence codante en une protéine requiert l'intervention de diverses protéines mentionnées précédemment : l'ARN-polymérase (responsable de la transcription de la séquence codante en ARN) et le ribosome sont les principaux, mais l'ARN-polymérase ne se lie à la molécule d'ADN que lorsque des facteurs de transcription sont eux-mêmes liés à la séquence dans le voisinage.

La fixation de chacune de ces molécules à la molécule d'ADN (ou d'ARN pour le ribosome) est due à la présence de motifs sur la séquence : un brin d'ARN qui ne porte pas la séquence de fixation du ribosome ne peut-être capturé par ce dernier, et n'est donc pas traduit. Il en est de même pour l'ARN-polymérase, dont la ligation à la molécule d'ADN est conditionnée par la présence d'un signal de fixation. Plus encore, ces motifs peuvent également induire un changement de conformation de la protéine effectrice de l'expression : c'est le cas des sites d'initiation et de terminaison de la traduction. Parallèlement, l'ARN-polymérase *reconnaît* un site d'*initiation de la transcription* et un site de terminaison de la transcription. Nous reviendrons sur les méthodes permettant la caractérisation, la représentation et la reconnaissance de ces signaux dans le chapitre suivant.

Mais au-delà de cette signature *extérieure* à la séquence de son caractère codant, il en existe également une signature intrinsèque à la séquence. Elle tient son origine dans un phénomène nommé *préférence du code*, et a été décrite pour la première fois en 1981 par SHEPHERD dans [**?**]. Il s'est inspiré de travaux dus à CRICK et ses collaborateurs qui précédèrent la découverte du code génétique, dans lesquels ils envisageaient un code génétique sans ponctuation (nous avons vu qu'au contraire, les régions codantes d'un

génome sont balisées, ponctuées, par les codons d'initiation et de terminaison de la transcription et de la traduction). Dans un tel système de codage, une question immédiate concerne la reconnaissance de la phase de lecture par la machinerie d'expression de la cellule : il faut pour cela que les codons *sensés* (ceux pour lesquels il existe un anticodon, rappelons que le véritable code génétique était alors inconnu) ne puissent être trouvés que sur une phase. Spéculant sur les conséquences d'une telle contrainte, et supposant que l'identification des phases de lecture s'appuie sur le schéma de succession des classes purine (R)/pyrimidine (Y), ils proposèrent le motif RRY comme définissant les triplets *valides* du code génétique. Ils fut établi plus tard que le motif RNY (N représentant les quatre nucléotides) pouvait être un bon candidat également, avec l'avantage sur le premier qu'il définit un ensemble de 16 codons valides au lieu de 8, beaucoup plus proches du nombre réel d'acides aminés différents (20) utilisés par le vivant.

SHEPHERD mit à profit ces observations pour construire une méthode de prédiction de la phase de lecture. Il analysa ainsi des fenêtres de 60 nucléotides, dénombrant le nombre minimal de mutations requises pour revenir à une séquence formée de 20 répétitions du motif RNY successives ; la phase de lecture nécessitant le plus petit nombre de mutations pour réaliser cette transformation était prédite comme phase de lecture d'une séquence codante se trouvant dans cette fenêtre. Les résultats étaient représentés graphiquement comme sur la figure 7, qui montre que les prédictions de la méthode sont pertinentes.

Les résultats de cette étude, pionnière dans l'analyse des séquences assistée par l'informatique, montrent qu'il y a effectivement un phénomène fort de *biais d'usage des codons* dans les séquences codantes. L'hypothèse de SHEPHERD est que historiquement, un code génétique sans ponctuation a précédé le code génétique actuellement utilisé par le vivant, et que les biais de composition exploités par sa méthode pour prédire les phases de lecture sont les traces évolutives de ce code. Les approches plus récentes de cette question se sont écartés de cette analyse évolutive, et se sont focalisées sur la quantification de ces phénomènes de préférence du code : pour chaque acide aminé, la distribution empirique d'occurence des différents codons synonymes dans les séquences codantes a été évaluée. Des biais de cette distribution ont pu être mis en évidence de la manière suivante : pour chaque acide aminé i, chaque codon j se voit affecté un poids $w_{i,j}$ égal au rapport de sa fréquence à celle du codon majoritaire parmi les synonymes codant également pour l'acide aminé i. L'*indice d'adapatation des codons* (ou CAI pour Codon Adaptation Index) d'un gène est alors défini comme la moyenne géométrique des poids des codons qui le composent. Des travaux plus récents (voir [**?**] et [**?**] pour plus de détails) ont montré que le CAI des différents gènes d'un même organisme peuvent varier grandement, et que les gènes de CAI élevé sont préférentiellement des gènes très conservés dans l'histoire évolutive de l'organisme.

Comme l'indique le titre de son article, SHEPHERD avait exploité le biais d'usage des codons pour prédire les positions et phases de lecture des régions codantes du génome. Nous verrons dans le chapitre suivant qu'il fut réellement un pionnier de l'analyse statistique des séquences génomiques, puisque la quasi-totalité des méthodes de détection de gène dans les génomes bactériens utilisent le biais d'usage des codons pour assigner une probabilité d'*être codante* à une région située entre un site d'initiation et un site de terminaison de la traduction et comprenant un nombre de nucléotides multiple de 3.

Par la suite, STADEN mena une analyse approfondie de la préférence du code ([**?**]), mais en dénombrant les triplets de nucléotides directement, sans référence aux catégories purine/pyrimidine. Il utilisa un modèle de fréquence des codons, $(f_{abc})_{(a,b,c)\in\mathcal{N}}$, ainsi qu'une distribution de la phase de lecture des séquences codantes, $(Q_1,\ldots,Q_3)$, qui revient à supposer qu'une proportion Q_i des codons est lue en phase i. Muni de ce

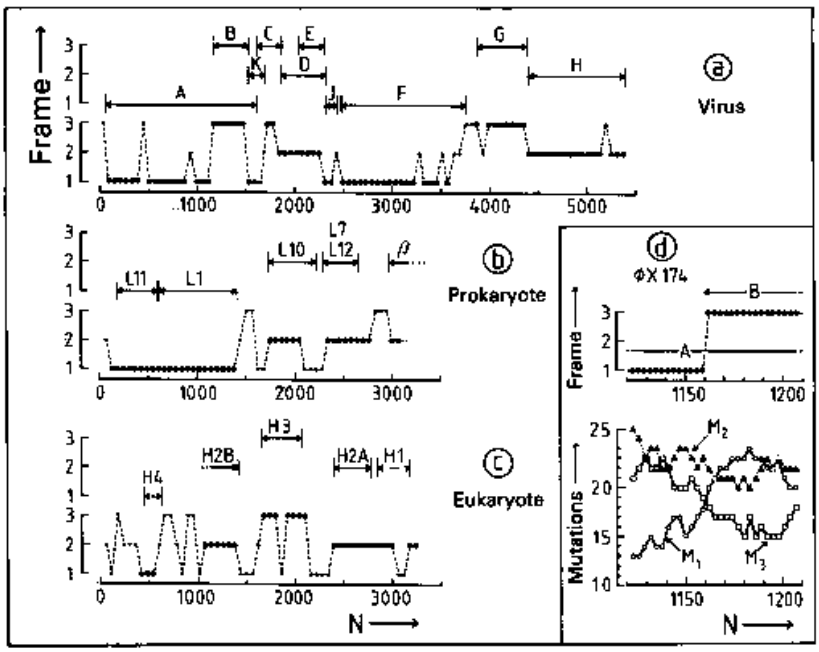


FIG. 7. Représentation graphique des résultats de prédiction des phases de lecture de SHEPHERD sur différentes portions de génome. En des positions de la séquence (abscisse) espacées de S nucléotides, l'ordonnée représente la phase de lecture assignée par la méthode pour la fenêtre de longueur L centrée en ce point. La partie haute de la figure représente les positions des gènes connus sur la séquence, ainsi que leur phase de lecture. (a) Virus $\phi X174$ avec $L = 60$ et une longueur de pas $S = 60$ (b) cluster de gènes ribosomaux du génome d'*Escherichia coli*, $L = 120$ et $S = 60$ (c) région en partie inconnue du génome de l'algue de mer, $L = 120$ et $S = 60$ (d) Analyse détaillée autour du site d'initiation du gène noté B dans la figure (a), $L = 60$ et $S = 3$. La figure est tirée de [**?**].

modèle, et en supposant que la séquence de longueur $3k$ est codante sur l'ensemble de sa longueur, la formule de Bayes permet de calculer la probabilité P_i que la séquence soit lue en phase i :

$$P_i = \frac{Q_i \exp H_i}{\sum_{j=1}^{3} Q_i \exp H_j}$$

où :

$$\begin{aligned} H_1 &= \sum_{i=1}^{k} \log f_{a_i b_i c_i} \\ H_2 &= \sum_{i=1}^{k} \log f_{b_i c_{i+1} a_{i+1}} \\ H_3 &= \sum_{i=1}^{k} \log f_{c_i a_{i+1} b_{i+1}} \end{aligned}$$

Bien entendu, une telle approche nécessite l'apprentissage des fréquences des codons dans de vraies séquences codantes pour lesquelles la phase de lecture est connue. Ce n'était en revanche pas le cas de la méthode précédente, qui postulait elle un motif

de succession des purines/pyrimidines le long d'une séquence codante lue en phase, et procédait par minimisation de l'écart entre la séquence et ce motif.

Pour en faire une méthode de prédiction des régions codantes dans une longue séquence d'ADN, STADEN calcule la probabilité de chacune des phases de lecture sur une fenêtre glissante dont la longueur est très inférieure à la longueur typique d'une région codante (au moins 600 nucléotides). La représentation des valeurs prises par cette probabilité en fonction de la position du milieu de la fenêtre permet de construire trois courbes de *probabilité de codage*; un pic persistant de l'une d'elles indique une région qui est probablement codante dans la phase de lecture correspondante.

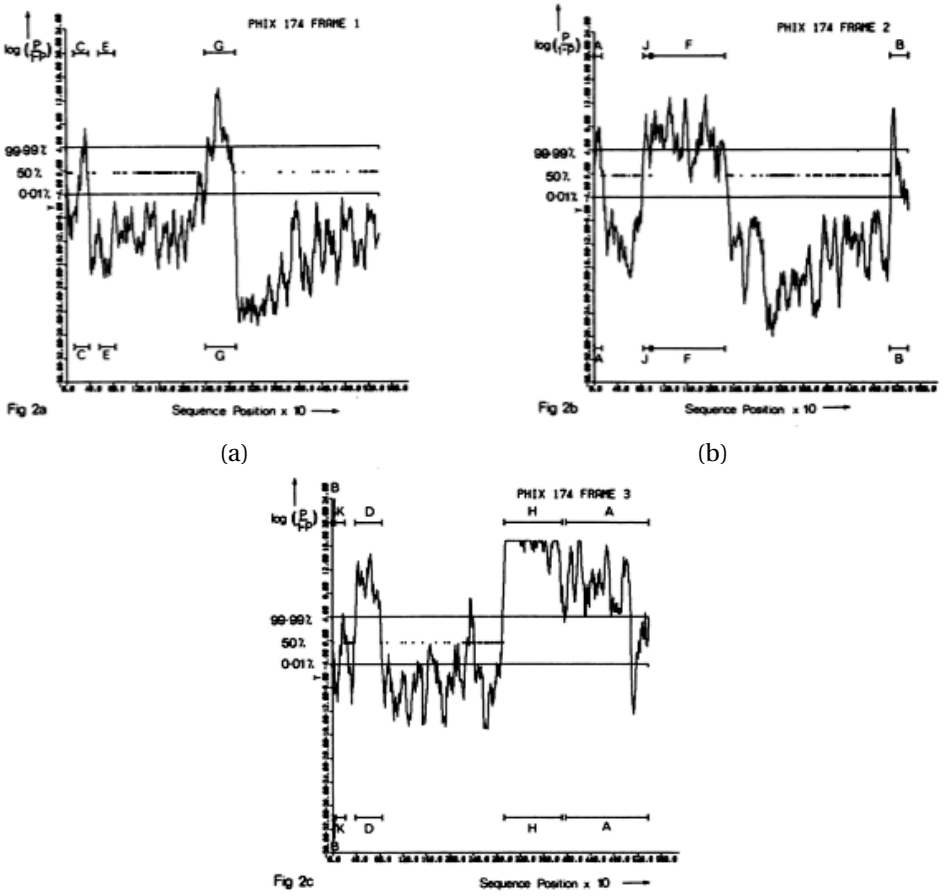


FIG. 8. Représentation graphique de l'information utilisée pour la détection des régions codantes et la détermination de leur phase.

3.2.4. *Autres types de signaux.* Nous concluons ce chapitre par la mention de quelques familles de signaux associés à d'autres aspects plus ou moins bien connus de la structure des génomes. Certains de ces signaux sont étroitement associés à l'expression des gènes : il en est ainsi des signaux d'épissage que sont les biais de composition de la séquence au voisinage d'un site où le brin d'ARN peut être coupé afin d'en retirer une portion. Les signaux environnants les sites d'épissage sont l'objet d'investigations approfondies à l'heure actuelle, la prédiction automatique de ces sites étant d'un grand intérêt pour la prédiction des gènes dans les organismes eucaryotes.

Il existe enfin des signaux qui sont associés à la compaction de l'ADN. La molécule d'ADN adopte en effet une conformation spatiale extrêmement compactes, résultant de

multiples repliements de celle-ci autour de protéines appelées *histones*. Comme toute interaction physico-chimique entre une protéine et une molécule d'ADN, elle est conditionnée par la composition de la séquence au voisinage du site de liaison. La recherche des régularités de la séquence associée à ce phénomène est également l'objet de recherches approfondies. Statistiquement, ces questions ne peuvent être abordées par les mêmes modélisations que la détection de gène par exemple. Il s'agit en effet de capturer des corrélations dont la portée est assez longue, contrairement aux séquences codantes qui présentent des biais de composition très locaux.

CHAPITRE 2

A la recherche des régularités

La plupart des propriétés de séquences présentées précédemment, concernant l'ADN comme les protéines, mettent l'accent sur des propriétés locales de la composition de la séquence : la présence d'une ORF, la conformité au code génétique dans les séquences codantes, la présence d'un signal dans une région de la séquence, la structure secondaire d'une protéine. Quantifier cette composition dans une région de la séquence passe naturellement par le dénombrement des mots de k-lettres, ou k-mots. Se pose alors la question du choix d'un modèle de la séquence en regard duquel ces comptages aient un sens. Nous proposons ici une construction de ce modèle fondé sur la maximisation de l'entropie, par une approche empruntée à la physique statistique, dont nous rappellerons la démarche et les résultats utiles pour la suite de l'exposé dans la première section. L'essentiel du contenu de l'exposé qui suit est une adaptation libre de textes dus principalement à E. T. JAYNES (voir [**?**], [**?**] et [**?**]). Il s'agit de définir le cadre de pensée qui nous guidera tout au long de ce texte.

1. L'approche de la physique statistique

La physique statistique s'intéresse à modéliser un système composé d'un grand nombre de particules en ne disposant que de mesures en moyenne. Il s'agit donc de modèles probabilistes, au sens où l'on décrit la distribution de probabilités qu'une particule soit dans un état particulier. Pour construire ces modèles, les physiciens appliquent en fait un principe d'incertitude, connu comme *le principe d'entropie maximale*. Ce principe peut être justifié de différentes manières : d'un point de vue philosophique premièrement, comme une conséquence d'un principe de rationnalité ; d'un point de vue opérationnel ensuite, au moyen d'un théorème de concentration de l'entropie que nous énoncerons et prouverons ; d'un point de vue de logique enfin, au travers du théorème de Cox que nous évoquerons plus loin dans ce manuscrit.

Notons $\mathscr{S}$ l'espace d'état des particules du système, et à chaque état $e \in \mathscr{E}$ possible, associons une ou plusieurs observables $E_1(e), \ldots, E_p(e)$. Pour le physicien, il s'agit typiquement du volume, de l'énergie, ou encore de l'entité chimique de la particule. Aucun caractère d'injectivité des fonctions $E_i : S \to \mathbb{R}$ n'est requis. Ces fonctions, qui prennent une valeur particulière pour chaque particule du système, sont inobservables ; ce qui est réellement observable, c'est le cumul (ou la moyenne, ce qui est équivalent lorsque le nombre de particules du système est constant) de ces quantités sur l'ensemble des n particules du système $(e_1, \ldots, e_n) \in \mathscr{E}^n$. On note ces quantités moyennes ou cumulées avec des crochets, $\langle E_i \rangle$, $i = 1, \ldots, p$. Ces quantités doivent nous servir à appréhender les états possibles du système formé des n particules, et en particulier à leur attribuer une mesure de plausibilité, autrement dit une probabilité. On s'intéresse donc maintenant aux configurations possibles du système S^n, et on note $\mathscr{D}$ l'ensemble des distributions de probabilité sur l'espace d'états $\mathscr{S}^n$.

Pour le statisticien, la perspective est un peu différente. L'espace d'états est le pendant de l'univers du probabiliste, et les observables, fonctions numériques de l'état, sont des variables aléatoires, qui définissent un espace image de l'univers. A cet espace image est associé une marginalisation de la distribution de probabilité sur l'univers. Si

les distributions d'état de deux systèmes diffèrent tout en présentant les mêmes distributions d'observables, les deux systèmes ne pourront être distingués. Elle a donc pour principale conséquence qu'une observation peut ne pas suffire à déterminer l'état du système de manière univoque. L'enjeu des applications des statistiques en sciences est précisément de prendre en compte l'incertitude qui persiste après l'observation.

1.1. Ensembles micro-canoniques. Il est d'ores et déjà possible de détailler les considérations combinatoires induites par les observations sur l'espace d'état $\mathscr{E}^n$ du système de n particules, que l'on appelle l'*espace des configurations*. En effet, chacune des configurations $\boldsymbol{e} = (e_1, \dots, e_n)$ du système donne lieu à l'observation :

$$\langle E_i \rangle = \frac{1}{n} \sum_{k=1}^{n} E_i(e_k), i = 1, \dots, p$$

et, en gébéral (ne serait-ce que par permutation des états si au moins deux d'entre eux diffèrent), plusieurs configurations peuvent donner lieu aux mêmes observations $\langle \boldsymbol{E} \rangle$. On est donc bien dans une situation de résolution d'un problème inverse qui admet plusieurs solutions. Ces solutions équivalentes, autrement dit les configurations du système qui donnent lieu aux mêmes observations moyennes, sont appelées les *ensembles micro-canoniques*.

DÉFINITION 1. *Soit un système de n particules dont l'espace de configuration est* $\mathscr{E}^n$*, et soit* $\langle \boldsymbol{E} \rangle$ *une observation moyenne d'une configuration. On appelle* ensemble micro-canonique *associé à l'observation* $\langle \boldsymbol{E} \rangle$ *l'ensemble* $\mathscr{M}(\langle \boldsymbol{E} \rangle)$ *l'ensemble des configurations qui donnent lieu à cette observation :*

$$\mathscr{M}(\langle \boldsymbol{E} \rangle) = \{\boldsymbol{e} \in \mathscr{E}^n, \frac{1}{n} \sum_{k=1}^{n} \boldsymbol{E}(e_k) = \langle E \rangle\}$$

Les ensembles micro-canoniques associés à un système forment une partition de l'espace des configurations du système. Chacune des parties de cette partition forme un ensemble de configurations *statistiquement indistinguables*.

1.2. Principe d'incertitude. Etant donnée une observation, $\langle \boldsymbol{E} \rangle = (E_1, \dots, E_p)$ effectuée sur un système constitué de n particules (on suppose ici que ce nombre n est connu), l'ensemble micro-canonique associé décrit donc l'ensemble des configurations compatibles avec cette observation. Mais au-delà de cette caractérisation ensembliste, qui présume une reproductibilité parfaite de l'observation moyenne, on peut s'intéresser à l'ensemble $\mathscr{D}(\langle \boldsymbol{E} \rangle)$ des distributions de probabilité sur l'espace des configurations qui prédisent effectivement cette observation. On entend par là que l'espérance des observables sous une distribution appartenant à $\mathscr{D}(\langle \boldsymbol{E} \rangle)$ coïncide avec leur moyenne observée :

$$\mathscr{D}(\langle \boldsymbol{E} \rangle) = \{\mathbb{P} \in \mathscr{D}, \mathbb{E}_{\mathbb{P}}(\boldsymbol{E}) = \langle \boldsymbol{E} \rangle\}$$

Chacune des distributions de l'ensemble $\mathscr{D}(\langle \boldsymbol{E} \rangle)$ définit un *macro-état* du système, au sens d'une description possible de la connaissance apportée sur le système par l'observation. Le terme macro-état exprime le fait que l'on ne décrit pas l'état de chacune des particules du système, mais plutôt un état macroscopique qui ne dicte que la distribution des états parmi les particules du système. Mais toutes les distributions de $\mathscr{D}(\langle \boldsymbol{E} \rangle)$ prédisent les observations, si bien qu'aucune contrainte réelle ne permet de restreindre cet ensemble de macro-états candidats. On peut d'ailleurs remarquer que la taille n du système n'intervient pas dans cette définition.

La question ici posée de choisir une explication parmi plusieurs possibles à un phénomène observé a occupé philosophes et scientifiques depuis des siècles. Selon JAYNES ([**?**]), on retrouve dans les travaux d'HÉRODOTE les prémices des principes que nous allons présenter maintenant. Mais parmi les discussions anciennes de cette question, le

*rasoir d'*OCCAM est l'une de celles auxquelles on se réfère le plus dans les textes modernes. OCCAM était un philosophe et logicien du $13^{ème}$ siècle, tenant du réductionnisme. Il avait posé que l'explication d'un phénomène devait requérir le moins d'hypothèses possibles, par l'élimination ou le *rasage* de celles qui ne changent pas la prédiction des observables donnée par la théorie explicative. D'où la maxime associée à OCCAM aujourd'hui : *les entités ne devraient pas être multipliées plus que nécessaire*.

Par la suite, les travaux d'OCCAM ont inspiré de nombreuses réflexions. Plusieurs siècles plus tard, PIERRE-SIMON DE LAPLACE énonça le principe d'incertitude introduit par OCCAM sous l'une de ses formes modernes. Le *principe de raison insuffisante*, qui dicte de donner une même probabilité à des événements qui peuvent être réalisés d'un même nombre de manière, établit un lien étroit entre le principe d'OCCAM et la théorie des probabilités. Ce principe affirme essentiellement qu'une théorie scientifique ne doit recourir qu'à l'ensemble d'hypothèses strictement nécessaires à ce qu'elle prédise les observables. En termes un peu plus abstraits, cela signifie qu'une théorie scientifique ne doit pas présenter plus de régularités[1] que celles requises par les observations. Dans le cadre probabiliste, il revient à choisir la distribution qui porte le moins d'information possible (ou, en d'autres termes, la distribution la moins prédictible) parmi celles qui prédisent l'observation (prédire l'observation signifiant que l'espérance est en accord avec les valeurs observées). On évite ainsi les prédictions parasites, puisqu'il n'est pas possible de choisir une distribution qui en permette moins sans violer la contrainte d'adéquation à la réalité.

La limite de l'énoncé précédent est avant tout pratique : il est nécessaire de recourir à une quantification de la régularité d'une distribution, ou, par opposition, de son désordre. Il s'est avéré depuis les travaux de LAPLACE qu'une telle quantification existe. Elle compte d'ailleurs parmi les objets scientifiques dont, historiquement, la maturation fut la plus longue, parmi les plus discutés aussi.

1.2.1. *L'entropie.* Cette quantité qui permet de mesurer le désordre d'une distribution, c'est l'*entropie*, dont la définition suit :

DÉFINITION 2. *L'entropie d'une distribution $\mathbb{P}$ sur un espace d'état dénombrable $\mathscr{E}$ s'écrit $H(\mathbb{P})$ et vaut :*

$$H(\mathbb{P}) = -\sum_{e\in\mathscr{E}} \mathbb{P}(e)\ln\mathbb{P}(e)$$

L'entropie a été introduite sous cette définition en 1948 par P. SHANNON dans [**?**] et [**?**]. Auparavant, cette quantité était cantonnée à la thermodynamique, cadre dans lequel elle avait été intuitée par CARNOT et CLAUSIUS, puis définie explicitement de manière restreinte par BOLTZMANN, qui ne considérait que le cas des distributions uniformes. En effet, si l'on considère la distribution uniforme sur tout l'espace d'états U, on a :

$$H(U) = |\mathscr{E}|\frac{1}{|\mathscr{E}|}\ln(|\mathscr{E}|) = \ln(|\mathscr{E}|)$$

ce qui, à constante multiplicative près, en l'occurence la constante de BOLTZMANN, coïncide avec l'expression que ce dernier avait donnée à cette quantité. Mais ce n'est pas la seule propriété de l'entropie, comme le montre le théorème suivant.

THÉORÈME 1. *L'entropie H est une fonctionnelle sur l'espace $\mathscr{D}$ des distributions de probabilité sur l'espace d'états $\mathscr{E}$ telle que :*

- $\forall\ \mathbb{P}\in\ \mathscr{D}, H(\mathbb{P})\geq 0$ *avec égalité si et seulement si* $\mathbb{P}$ *est une distribution de Dirac.*
- $\forall\ \mathbb{P}\in\mathscr{D}, H(\mathbb{P})\leq\ln(|\mathscr{E}|)$ *avec égalité si et seulement si* $\mathbb{P}$ *est la distribution uniforme sur* $\mathscr{E}$.

[1] Régularité signifie ici prédictions : la théorie ne doit rien prédire de plus que ce que l'observation de la réalité impose qu'elle prédise.

La première assertion tient au fait que $\ln \mathbb{P}(e) \leq 0$ pour tout $a \in \mathscr{E}$, puisque $\mathbb{P}(e) \leq 1$. Par conséquent, l'entropie est l'opposé d'une moyenne de termes négatifs. Elle est donc positive ou nulle. Il est immédiat de constater qu'une distribution de Dirac a une entropie nulle. De plus, si aucun des termes $\mathbb{P}(e)$ ne vaut 1, alors il existe au moins un terme $\mathbb{P}(e)$ compris entre 0 et 1 strictement, si bien que $H(\mathbb{P}) \geq -\mathbb{P}(e)\ln \mathbb{P}(e) > 0$, ce qui conclut le cas d'égalité.

La seconde assertion découle d'une optimisation sous la contrainte de rester dans l'espace des distributions de probabilité $\mathscr{D}(\mathscr{E})$:

$$\begin{aligned} &\max H(\mathbb{P}) \\ s.c. &\sum_{e\in\mathscr{E}} \mathbb{P}(e) = 1 \end{aligned}$$

Le lagrangien de ce problème d'optimisation s'écrit :

$$\mathscr{L}(\mathbb{P}, \lambda) = H(\mathbb{P}) - \lambda(\sum_{e\in\mathscr{E}} \mathbb{P}(e) - 1)$$

ce qui conduit au système différentiel d'optimalité suivant :

$$\frac{\partial \mathscr{L}}{\partial \mathbb{P}(e)} = -1 - \ln \mathbb{P}(e) - \lambda = 0, \quad \forall\, e \in \mathscr{E} \tag{1}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda} = \sum_{e\in\mathscr{E}} \mathbb{P}(e) - 1 = 0 \tag{2}$$

ce qui conduit à la solution :

$$\mathbb{P}(e) = \exp 1 + \lambda, \quad \forall\, e \in \mathscr{E}$$

qui coïncide bien avec la distribution uniforme. Par ailleurs, cette solution est unique. Ce qui conclut le second point. □

Ainsi, l'entropie est une fonctionnelle qui est minimale pour les distributions déterministes et maximale pour la distribution uniforme. Plus généralement, cette quantité mesure le *désordre* d'une distribution de probabilité : plus elle est grande, moins l'issue d'un tirage dans cette distribution est prédictible. Nous verrons plus loin une interprétation de cette quantité en terme de la plus faible longueur de code moyenne atteignable par une méthode de compression sans perte d'une série de tirages de la distribution.

1.2.2. *Principe d'entropie maximale.* Mais revenons au principe d'incertitude. Si l'on retient l'entropie comme mesure du désordre d'une distribution, il devient le principe d'entropie maximale, dont la formulation suit.

PRINCIPE D'ENTROPIE MAXIMALE. *Parmi l'ensemble des distributions qui prédisent les observations, il faut toujours choisir celle dont l'entropie est la plus grande.*

Suivant ce principe, la distribution de probabilité dans l'ensemble $\mathscr{D}(\langle \boldsymbol{E} \rangle)$ retenue pour décrire le système doit être la solution du problème de maximisation suivant :

$$\begin{aligned} &\max && H(\mathbb{P}) \\ &s.c. && \begin{cases} \sum_{\boldsymbol{e}\in\mathscr{E}^n} \mathbb{P}(\boldsymbol{e}) = 1 \\ \mathbb{E}_{\mathbb{P}}(\boldsymbol{E}) = \langle \boldsymbol{E} \rangle \end{cases} \end{aligned} \tag{3}$$

Le lagrangien associé à ce nouveau problème d'optimisation (3) est :

$$\mathscr{L}(\mathbb{P}, \lambda, \mu) = H(\mathbb{P}) - \lambda(\sum_{\boldsymbol{e}\in\mathscr{E}^n} \mathbb{P}(\boldsymbol{e}) - 1) - \sum_{i=1}^{p} \mu_i(\mathbb{E}_{\mathbb{P}}(E_i) - \langle E_i \rangle)$$

où λ est le multiplicateur de Lagrange associé à la première contrainte, $\sum_{\boldsymbol{e}\in\mathscr{E}^n} \mathbb{P}(\boldsymbol{e}) = 1$, et μ_i celui associé à la contrainte portant sur la moyenne de l'observable i, $\mathbb{E}_{\mathbb{P}}(E_i) = \langle E_i \rangle$.

Le système d'optimalité s'écrit alors :

$$\frac{\partial \mathscr{L}}{\partial \mathbb{P}(\boldsymbol{e})} = -1 - \ln \mathbb{P}(\boldsymbol{e}) - \lambda - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e}) = 0, \quad \forall\, \boldsymbol{e} \in \mathscr{E}^n \tag{4}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda} = -\sum_{\boldsymbol{e}\in\mathscr{E}^n} \mathbb{P}(\boldsymbol{e}) + 1 = 0 \tag{5}$$

$$\frac{\partial \mathscr{L}}{\partial \mu_i} = \langle E_i \rangle - \mathbb{E}_{\mathbb{P}}(E_i) = 0 \tag{6}$$

dont les solutions sont de la forme :

$$\mathbb{P}(\boldsymbol{e}) = \exp -1 - \lambda - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e}), \quad \forall \boldsymbol{e} \in \mathscr{E}^n$$

Par une translation d'une unité du multiplicateur de Lagrange λ, on obtient :

$$\mathbb{P}(\boldsymbol{e}) = \frac{1}{e^{\lambda}} \exp - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e}), \quad \forall \boldsymbol{e} \in \mathscr{E}^n$$

Comme on doit avoir $\sum_{\boldsymbol{e}\in\mathscr{E}^n} \mathbb{P}(\boldsymbol{e}) = 1$, on a par conséquent :

$$\lambda = \ln \sum_{\boldsymbol{e}\in\mathscr{E}^n} \exp - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e})$$

Il est usuel de noter $Z(\boldsymbol{\mu}) = \sum\limits_{\boldsymbol{e}\in\mathscr{E}^n} \exp - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e})$, que l'on appelle la *fonction de partition*, si bien que l'on a :

$$\mathbb{P}(\boldsymbol{e}) = \frac{1}{Z(\boldsymbol{\mu})} \exp - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e}), \quad \forall \boldsymbol{e} \in \mathscr{E}^n \tag{7}$$

Il nous reste à présent à calculer les *variables conjuguées* μ en fonction des observations moyennes. Il suffit pour cela de remarquer que :

$$\frac{\partial \ln Z}{\partial \mu_i} = \frac{1}{Z(\boldsymbol{\mu})} \sum_{\boldsymbol{e}\in\mathscr{E}^n} -E_i(\boldsymbol{e}) \exp - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e})$$

ce qui montre que :

$$\mathbb{E}_{\boldsymbol{\mu}}(E_i) = -\frac{\partial \ln Z}{\partial \mu_i}, \quad i = 1, \ldots, p$$

et la contrainte d'appartenance de la distribution à $\mathscr{D}(\langle \boldsymbol{E} \rangle)$ impose la valeur de μ telle que :

$$\frac{\partial \ln Z}{\partial \mu_i} = -\langle E_i \rangle, \quad i = 1, \ldots, p$$

Nous pouvons à présent, pour plus de commodité, résumer ces résultats dans un unique théorème.

THÉORÈME 2. *La distribution dont l'entropie est maximale parmi les distributions de* $\mathscr{D}(\langle \boldsymbol{E} \rangle)$ *est de la forme :*

$$\mathbb{P}(\boldsymbol{e}) = \frac{1}{Z(\boldsymbol{\mu})} \exp - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e}) \tag{8}$$

où $Z(\boldsymbol{\mu}) = \sum_{\boldsymbol{e}\in\mathscr{E}^n} \exp - \sum_{i=1}^{p} \mu_i E_i(\boldsymbol{e})$ *et* $\boldsymbol{\mu}$ *tel que :*

$$\frac{\partial \ln Z}{\partial \mu_i} = -\langle E_i \rangle, \quad i = 1, \ldots, p \tag{9}$$

Remarquons qu'un modèle exponentiel sur un espace d'état discret est classiquement défini comme un ensemble de distributions de probabilités $(\mathbb{P}_{\boldsymbol{\theta}})_{\boldsymbol{\theta}\in\Theta}$ indexé par un domaine Θ de $\mathbb{R}^p$ telles qu'il existe une moyenne observable $\boldsymbol{E} : \mathscr{E}^n \to \mathbb{R}^p$ et une fonction de masse $\lambda : \Theta \to \mathbb{R}_+$ pour lesquelles :

$$\forall \theta \in \Theta, \forall \boldsymbol{e} \in \mathscr{E}^n, \mathbb{P}_\theta(\boldsymbol{e}) = \frac{1}{\lambda(\theta)} \exp - \sum_{i=1}^{p} \theta_i E_i(\boldsymbol{e})$$

Le théorème 2 fournit donc une justification du recours presque systématique à ce type de modèle : celui-ci apparaît en effet naturellement comme la solution du maximum d'entropie sur l'ensemble des distributions qui prédisent les observations moyennes. Le rôle central de la moyenne dans l'estimation de ce modèle (cette dernière est en effet une statistique exhaustive minimale dans un modèle exponentiel) s'en trouve également justifié : par construction, un modèle exponentiel est paramétré de manière bijective par la moyenne de ses observables.

1.2.3. *Concentration de l'entropie.* Au-delà de la justification quelque peu philosophique du recours au principe de maximum d'entropie, ce dernier admet également une justification opérationnelle. En effet, l'entropie possède la propriété de se *concentrer* avec la taille de l'échantillon : les configurations dont les distributions empiriques sont proches de celle maximisant l'entropie occupent une proportion d'autant plus grande parmi les configurations compatibles avec les contraintes observées que la taille de l'échantillon est grand.

THÉORÈME 3. *Soit $\langle \boldsymbol{E} \rangle$ une observation moyenne sur n particules, et notons $\mathbb{P}$ la distribution d'indépendance associée à $\mathbb{P}^\star_{\langle E \rangle}$: $\mathbb{P}^{\otimes n} = \mathbb{P}^\star_{\langle E \rangle}$. Si l'on note $\mathbb{U}$ la distribution uniforme sur l'espace d'états $\mathscr{S}^n$, on a :*

$$\forall \varepsilon > 0, U^{\otimes n}(\sup_{s\in\mathscr{S}} |\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{x_i = s\}} - \mathbb{P}(s)| > \varepsilon | \langle \boldsymbol{E} \rangle) = O(\exp - c_\varepsilon n)$$

où c_ε est une constante strictement positive dépendant de ε.

La preuve de ce théorème peut être trouvée dans [**?**].

Plus prosaïquement, ce résultat établit qu'asymptotiquement, seul le micro-canonique maximisant l'entropie sous la contrainte observée se réalise, avec une probabilité d'autant plus proche de 1 que la taille de l'échantillon est élevée. Et que la distribution de maximum d'entropie converge étroitement vers la distribution uniforme sur le micro-canonique. Ce dernier point fait le lien avec le conception de l'entropie introduite par BOLTZMANN.

En pratique, ce principe est d'une importance notoire. Puisque les configurations dont la distribution empirique des états diffèrent de la distribution de maximum d'entropie sont en proportion exponentiellement décroissante avec la taille de l'échantillon, cela signifie que, pour n'importe quelle fonction $\boldsymbol{f} : \mathscr{S} \to \mathbb{R}^p$ de l'état des particules, on a également le résultat suivant.

COROLLAIRE 3.1. *Soit $\langle \boldsymbol{E} \rangle$ une observation moyenne sur n particules. La moyenne empirique de la fonction $\boldsymbol{f}$ sur une configuration converge vers $\mathbb{E}_{\mathbb{P}}(\boldsymbol{f})$ avec n, au sens où :*

$$\forall \varepsilon > 0, \mathbb{U}^{\otimes n}(\sup_{k=1}^{p} |\mathbb{E}_{\mathbb{P}}(f_k) - \frac{1}{n}\sum_{i=1}^{n} f(x_i)| > \varepsilon | \langle \boldsymbol{E} \rangle) = O(\exp - c'_\varepsilon n)$$

La preuve est immédiate puisque l'espace d'état est supposé fini, donc f est bornée. □

Remarquons que ce résultat est un résultat purement combinatoire : il s'appuie sur la mesure uniforme, et donc concerne fondamentalement des *nombres* de configurations. Ici, rien ne préjuge d'une *véritable* distribution des états. D'ailleurs, ceux-ci peuvent tout à fait suivre une autre distribution : mais dans ce cas, ce sont les choix des observables qui ne permettent pas d'accéder à l'ensemble des régularités du système.

1.3. Conséquences du principe d'entropie maximale.

1.3.1. *Calcul de l'entropie maximale.* Par le recours à la méthode de Lagrange pour optimiser l'entropie de la distribution sous la contrainte de prédire les observations, nous avons introduit un système de coordonnées alternatif à celui des espérances pour paramétrer l'espace des distributions solutions du problème d'optimisation. Nous allons voir que ces systèmes de coordonnées entretiennent des relations étroites.

La principale relation qui les relie apparaît en répondant à la première question naturelle après avoir mis en œuvre le principe de maximum d'entropie à partir d'observations moyennes : quelle est la valeur de l'entropie obtenue ? Il apparaît que son expression est extrêmement simple.

THÉORÈME 4. *La valeur $H^\star(\langle \boldsymbol{E} \rangle)$ de l'entropie au maximum d'entropie contraint par les observations $\langle \boldsymbol{E} \rangle$ est :*

$$H^\star(\langle \boldsymbol{E} \rangle) = \ln Z(\mu) + \sum_{i=1}^{p} \mu_i E_i \tag{10}$$

La preuve de ce théorème tient à un simple calcul. En effet :

$$H^\star(\langle \boldsymbol{E} \rangle) = -\sum_{\boldsymbol{e} \in \mathscr{E}^n} \mathbb{P}(\boldsymbol{e}) \ln \mathbb{P}(\boldsymbol{e})$$

où $\mathbb{P}$ désigne la distribution d'entropie maximale dans $\mathscr{D}(\langle \boldsymbol{E} \rangle)$. Il vient donc d'après le théorème 2 :

$$H^\star(\langle \boldsymbol{E} \rangle) = \ln Z(\mu) \sum_{\boldsymbol{e} \in \mathscr{E}^n} \mathbb{P}(\boldsymbol{e}) + \sum_{i=1}^{p} \mu_i \sum_{\boldsymbol{e} \in \mathscr{E}^n} \mathbb{P}(\boldsymbol{e}) E_i(\boldsymbol{e})$$

soit encore :

$$H^\star(\langle \boldsymbol{E} \rangle) = \ln Z(\mu) + \sum_{i=1}^{p} \mu_i \langle E_i \rangle$$

ce qui établit le résultat annoncé. □

La relation (10) met en jeu les deux système de coordonnées introduits sur l'ensemble $\mathscr{D}^\star$ des distributions qui maximisent l'entropie pour une valeur quelconque des observations :

- les *coordonnées moyennes*, ou *en espérance*, $\langle \boldsymbol{E} \rangle$, qui sont bien en bijection avec l'ensemble des distributions dans $\mathscr{D}^\star$ par existence et unicité de la distribution solution de (3) ;
- les *coordonnées lagrangiennes*, ou *conjuguées*, μ, qui définissent de manière unique une distribution à travers la formule (7) et la contrainte $\sum_{a \in \mathscr{A}} \mathbb{P}(a) = 1$.

Ces deux systèmes de coordonnées sont liés par la relation :

$$\langle E_i \rangle = -\frac{\partial \ln Z}{\partial \mu_i}, \quad i = 1, \ldots, p$$

La formule (10) exprime cette relation différemment, en montrant que l'entropie et le logarithme de la fonction de partition sont les transformées de LEGENDRE l'une de l'autre.

1.3.2. *Uniformité sur les ensembles micro-canoniques.* L'équation (8) qui fournit la solution explicite au problème de maximum d'entropie présente une propriété particulière : la probabilité d'une configuration du système ne dépend que de ses niveaux d'énergie cumulés. Autrement dit, toutes les configurations présentant les mêmes niveaux d'énergie cumulés ont une même probabilité. Cette remarque se formule sous la forme du résultat suivant :

THÉORÈME 5. *Toute distribution de maximum d'entropie sur l'espace des configurations est uniforme sur chacun des ensembles micro-canoniques.*

Par conséquent, la probabilité de toute configuration peut-être décomposée sous la forme d'un produit de deux probabilités, celle du microcanonique et celle de l'élément conditionnellement à son appartenance au micro-canonique :

$$\forall \boldsymbol{e} \in \mathscr{M}(\langle E \rangle), \mathbb{P}(\boldsymbol{e}) = \mathbb{P}(\mathscr{M}(\langle \boldsymbol{E} \rangle)) \times \frac{1}{|\mathscr{M}(\langle \boldsymbol{E} \rangle)|}$$

En d'autres termes, le principe de maximum d'entropie assigne des probabilités aux ensemble micro-canoniques, lesquelles se traduisent en probabilités sur les configurations au travers du théorème 5.

1.3.3. *Indépendance des particules.* Si maintenant on revient à la définition de $E_i(\boldsymbol{e})$ comme le cumul des énergies des particules dans la configuration $\boldsymbol{e}$, i.e. :

$$\forall i = 1, \ldots, p, \forall \boldsymbol{e} \in \mathscr{E}^n, E_i(\boldsymbol{e}) = \sum_{k=1}^{n} E_i(e_k)$$

la distribution de maximum d'entropie s'écrit :

$$\forall \boldsymbol{e} \in \mathscr{E}^n, \mathbb{P}(\boldsymbol{e}) = \frac{1}{Z} \exp - \sum_{i=1}^{p} \sum_{k=1}^{n} \mu_i E_i(e_k)$$

soit encore :

$$\mathbb{P}(\boldsymbol{e}) = \frac{1}{Z} \prod_{k=1}^{n} \exp - \sum_{i=1}^{p} \mu_i E_i(e_k)$$

On reconnaît dans cette relation l'indépendance des particules constituant le système, ce qui prouve le théorème suivant.

THÉORÈME 6. *Lorsque les énergies observées sont les moyennes sur l'ensemble des particules d'énergies individuelles, la distribution sur les configurations maximisant l'entropie sous la contrainte de prédire les moyennes observées est une distribution sous laquelle les particules du système sont indépendantes.*

Par conséquent, à la distribution $\mathbb{P}$ sur $\mathscr{E}^n$ obtenue par maximum d'entropie correspond une distribution sur $\mathscr{E}$, que nous noterons $\hat{\mathbb{P}}$, et qui est telle que :

$$\forall e \in \mathbb{E}, \mathbb{P}(e) = \frac{1}{Z(\boldsymbol{\mu})} \exp - \sum_{i=1}^{p} \mu_i E_i(e)$$

ou, de manière plus compacte, telle que $\mathbb{P} = \hat{\mathbb{P}}^{\otimes n}$. Parmi les distributions qui assurent l'indépendance des particules constituant le système, celle-ci est la seule qui prédise les moyennes observées. Elle ne résulte donc pas de la maximisation à proprement parler, mais des contraintes qui lui sont imposées : ce que le principe de maximum d'entropie affirme fondamentalement, c'est que $\mathbb{P} = \hat{\mathbb{P}}^{\otimes n}$.

2. Les chaînes de Markov

Nous allons voir à présent comment l'application du principe de maximum d'entropie au cas des séquences permet de reconstruire le modèle sous-jacent à l'ensemble des méthodes de modélisation de séquences que nous exposerons dans le prochain chapitre : les chaînes de Markov. Par le terme *séquence*, on entend une suite finie $\boldsymbol{x} = (x_1, \ldots, x_n)$ de symboles pris dans un alphabet $\mathscr{A}$ fini. Suivant la démarche exposée précédemment, nous définissons dans un premier temps l'espace d'état du système, puis les ensembles microcanoniques, pour enfin résoudre le problème de maximisation de l'entropie sous la contrainte de prédire les observations choisies.

2.1. L'espace d'états et les observables.

2.1.1. *Notations.* Le caractère ordonné des séquences leur confère une structure d'ordre supplémentaire qui les distingue d'un simple échantillon de même taille n : tous les symboles qui composent la séquence ont chacun deux *voisins* (à l'exception du premier symbole de la séquence et du dernier), leur prédécesseur dans la séquence et leur successeur. On peut en particulier parler de k-*mots*, au sens de sous-suites formées de k symboles consécutifs dans la séquence. Afin de faciliter l'exposé, nous adopterons dans la suite la notation $\boldsymbol{x}_{i,i+k}$ pour désigner la sous-suite $(x_i,\dots,x_{i+k})$ de la séquence $\boldsymbol{x}$.

2.1.2. *Particules et espace d'états.* Pour prolonger le parallèle avec l'approche introduite dans le cadre de la physique statistique, il semble naturel de considérer chaque lettre composant la séquence comme une particule du système. Cependant, si l'on se restreint au comptage des lettres, on retombe immédiatement sur le cas précédent d'échantillons non-structurés, et cela ne permet donc pas de prendre en compte utilement la structure d'ordre dont la séquence est munie alors même qu'il s'agit là de l'objectif de ce travail de construction du modèle. On souhaite en effet disposer d'un modèle qui permette de mettre à profit, par exemple, les observations de SHEPHERD quant à la composition en mots de 3 lettres des séquences codantes.

Aussi le dénombrement des k-mots, avec $k>1$, fournit-il une observation des interactions entre k particules successives qui était absente du développement précédent sur le principe de maximum d'entropie. Cependant, cette approche permet de construire un modèle adapté à ce nouveau cadre. Dans un premier temps, et afin de mettre en évidence la manière dont le principe de maximum d'entropie permet de modéliser des séquences présentant des dépendances, nous nous restreindrons au cas $k=2$, puis étendrons les résultats au cas $k>2$.

2.1.3. *Espace des configurations et ensembles micro-canoniques.* L'espace de configuration des séquences de longueur donnée n sur un alphabet $\mathscr{A}$ est évidemment $\mathscr{A}^n$. Mais du fait de l'observation des comptage des mots de 2 lettres dans la séquence, cet ensemble se trouve partitionné par les observations en ensembles micro-canoniques différents du cas indépendant développé précédemment.

Soit donc une observation $\boldsymbol{N}=(N_{a_1a_2})_{(a_1,a_2)\in\mathscr{A}^2}$ des comptages des mots de 1 et 2 lettres. L'ensemble micro-canonique associé s'écrit :

$$\mathscr{M}(\boldsymbol{N})=\{\boldsymbol{x}\in\mathscr{A}^n,\forall(a_1,a_2)\in\mathscr{A}^2,N_{a_1a_2}(\boldsymbol{x})=N_{a_1a_2},N_{a_1}(\boldsymbol{x})=N_{a_1}\}$$

En fait, si l'on prend en compte la nature de la première lettre de la séquence (ou de la dernière), les comptages de chacune des lettres se déduisent de ceux des mots de deux lettres :

$$\forall a\in\mathscr{A},\forall\boldsymbol{x}\in\mathscr{A}^n,N_a(\boldsymbol{x})=\delta_{x_n,a}+\sum_{a'\in\mathscr{A}}N_{aa'}(\boldsymbol{x})=\delta_{x_1,a}+\sum_{a'\in\mathscr{A}}N_{a'a}(\boldsymbol{x})$$

où $\delta_{a,a'}$ vaut 1 si $a=a'$, et 0 sinon. Autrement formulée, cette remarque signifie que la connaissance des comptages des mots de 1 et 2 lettres est équivalente à celle des comptages des mots de 2 lettres, ainsi que de la première lettre de la séquence. Remarquons enfin qu'un ensemble micro-canonique est ce que l'on appelle, dans le cadre de la théorie de l'information, un *type* [**?**].

La détermination des cardinaux des ensembles micro-canoniques a été mené par WHITTLE[**?**]. Celui-ci s'intéressait déjà aux problématiques de compression de texte, et s'interrogeait sur l'opportunité du schéma de compression suivant : on transmet d'abord les comptages des mots de 2 lettres, puis la première lettre, puis un *numéro d'ordre* établi préalablement (par exemple en utilisant l'ordre alphabétique) de la séquence à transmettre parmi celle qui commencent effectivement par la première lettre indiquée, et qui ont les nombres d'occurrence de chacun des mots de 2 lettres spécifiés. La compression atteinte par un tel schéma de codage dépend fondamentalement du nombre de bits nécessaire pour encoder le fameux numéro d'ordre de la séquence au sein de son micro-canonique, et par suite du nombre de séquences présentes dans un

micro-canonique particulier. Cela l'a conduit à établir la formule éponyme, qui est résumée par le théorème suivant. On considère, pour les besoins de l'exposé, que l'alphabet est ordonné sous la forme $\mathscr{A} = \{a_1, \ldots, a_p\}$, et que les comptages des mots de deux lettres sont formulés sous la forme d'une matrice $\boldsymbol{N}$ de taille $|\mathscr{A}| \times |\mathscr{A}|$, où $N_{i,j} = N_{a_i a_j}$.

THÉORÈME 7. *Soit N une matrice de comptage des mots de 2 lettres, et soit $\mathscr{M}(\boldsymbol{N})$ l'ensemble micro-canonique associé. Alors :*

$$|\mathscr{M}(\boldsymbol{N})| = K(\boldsymbol{N}) \sum_{i,j=1}^{p} H_{i,j}(\boldsymbol{N})$$

où :

$$K(\boldsymbol{N}) = \frac{\prod_{i=1}^{p} N_{a_i}!}{\prod_{i,j=1}^{p} N_{i,j}!}$$

et $H_{i,j}(\boldsymbol{N})$ désigne le cofacteur (i,j) de la matrice $\mathbb{I}_p - \bar{\boldsymbol{N}}$ définie par :

$$\forall\, i,j \in \{1,\ldots,p\}, \bar{N}_{i,j} = \frac{N_{i,j}}{\sum_{k=1}^{n} N_{i,k}}$$

La preuve de ce théorème peut être trouvée dans [**?**].

2.2. Evaluation de la fonction de partition. Calculons formellement la fonction de partition. Elle est définie comme la somme sur les configurations $\boldsymbol{x}$ possibles des quantités $\exp -\sum_{i=1}^{p} \mu_i E_i(\boldsymbol{x})$, soit :

$$\ln Z(\boldsymbol{\mu}) = \sum_{\boldsymbol{x} \in \mathscr{A}_k^n} \exp\, - \sum_{t=1}^{n-1} \mu_{x_t, x_{t+1}}$$

Cette expression, lourde et difficilement manipulable, possède un équivalent matriciel qui permet une simplification drastique des calculs. Nous introduisons pour cela la *matrice de transfert* du système associée au paramètre $\boldsymbol{\mu}$, M_μ, de dimension $|\mathscr{A}| \times |\mathscr{A}|$, et définie par la relation :

$$\forall\, a, a' \in \mathscr{A}, M_{\boldsymbol{\mu}}(a, a') = \exp -\mu_{aa'}$$

Le théorème suivant énonce la formulation équivalente de la fonction de partition utilisant la matrice de transition.

THÉORÈME 8. *La fonction de partition s'écrit, en fonction de la matrice de transfert, de la manière suivante :*

$$Z_n(\boldsymbol{\mu}) = \sum_{x_1, x_n \in \mathscr{A}} M_\mu^n(x_1, x_n) \tag{11}$$

La preuve de ce résultat s'obtient par une simple récurrence sur la longueur n de la séquence. On utilisera plutôt la notation en termes de lettres (et non de mots), plus manipulable pour ce calcul. La récurrence s'initialise dans le cas $n = 2$ (longueur requise pour observer deux lettres consécutives), pour lequel l'identité (11) s'écrit :

$$\sum_{\boldsymbol{x} \in \mathscr{A}^2} \exp -\mu(x_1, x_2) = \sum_{x \in \mathscr{A}^2} M_\mu(x_1, x_2)$$

qui est vérifiée compte-tenu de la définition de M_μ. Supposons donc la relation établie pour des séquences de longueur inférieure à n, et regroupons les termes dans l'expression de la fonction de partition :

$$Z_{n+1}(\boldsymbol{\mu}) = \sum_{x_{n+1} \in \mathscr{A}} \sum_{x_1 \in \mathscr{A}} \sum_{x_n \in \mathscr{A}} M_\mu^n(x_1, x_n) \exp -\mu(x_n, x_{n+1})$$

Compte-tenu de la définition de la matrice M, il vient :

$$Z_{n+1}(\boldsymbol{\mu}) = \sum_{x_{n+1} \in \mathscr{A}} \sum_{x_1 \in \mathscr{A}} \sum_{x_n \in \mathscr{A}} M_\mu^n(x_1, x_n) M_\mu(x_n, x_{n+1})$$

On reconnaît alors la dernière somme comme un produit matriciel :

$$Z_{n+1}(\boldsymbol{\mu}) = \sum_{x_{n+1}\in\mathscr{A}} \sum_{x_1\in\mathscr{A}} M_\mu^{n+1}(x_1, x_{n+1})$$

ce qui conclut la preuve par récurrence. □

2.3. Distribution de maximum d'entropie. D'après le théorème 2, la distribution $\mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}$ sur l'espace de configuration des séquences $\mathscr{A}^n$ maximisant l'entropie sous la contrainte de prédire les comptages des mots de une ou deux lettres $\langle \boldsymbol{N}\rangle$ vérifie :

$$\forall \boldsymbol{x}\in\mathscr{A}^n, \mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}(\boldsymbol{x}) = \frac{1}{Z_n(\mu)} \exp - \sum_{a\in\mathscr{A}} N_a(\boldsymbol{x})\mu_a - \sum_{(a_1a_2)\in\mathscr{A}^2} N_{a_1a_2}(\boldsymbol{x})\mu_{a_1a_2}$$

où $(\mu_a)_{a\in\mathscr{A}}$ désigne les multiplicateurs de LAGRANGE associés aux comptages des lettres, et $(\mu_{a_1a_2})_{(a_1,a_2)\in\mathscr{A}^2}$ ceux associés aux comptages des mots de 2 lettres.

Dans le cas de l'observation des fréquences des états des particules individuelles, nous avons vu que le principe de maximum d'entropie assignait à toute observation une distribution sous laquelle les particules sont indépendantes. Dans le cas présent, cette propriété n'est plus vérifiée, et la propriété de *Markov* s'y substitue.

DÉFINITION 3. *Une distribution $\mathbb{P}$ sur l'espace de configuration $\mathscr{A}^n$ des séquence de longueur n sur un alphabet fini $\mathscr{A}$ possède la propriété de Markov si :*

$$\forall t > 1, \forall \boldsymbol{y}\in\mathscr{A}^t$$

$$\mathbb{P}(\{\boldsymbol{x}, x_{1,t} = y_{1,t}\}|\{\boldsymbol{x}, x_{1,t-1} = y_{1,t-1}\}) = \mathbb{P}(\{\boldsymbol{x}, x_{t-1,t} = y_{t-1,t}\}|\{\boldsymbol{x}, \boldsymbol{x}_{t-1} = y_{t-1}\}) \tag{12}$$

Cette propriété caractérise les distributions sur les configurations des séquences maximisant l'entropie, et ce quelques soient les valeurs des contraintes de comptages dont elle sont issues.

THÉORÈME 9. *Quelque soit le vecteur de comptage observé $\langle \boldsymbol{N}\rangle$, la distribution $\mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}$ vérifie la propriété de* MARKOV.

En effet, soit $\boldsymbol{y}\in\mathscr{A}^t$ une séquence de t lettres. Le membre de gauche de l'équation (12) s'écrit :

$$\mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}(\{\boldsymbol{x}, \boldsymbol{x}_{1,t} = \boldsymbol{y}_{1,t}\}|\{\boldsymbol{x}, \boldsymbol{x}_{1,t-1} = \boldsymbol{y}_{1,t-1}\}) = \frac{\sum_{\boldsymbol{x}\in\mathscr{A}^n, \boldsymbol{x}_{1,t}=\boldsymbol{y}_{1,t}} \mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}(\boldsymbol{x})}{\sum_{\boldsymbol{x}\in\mathscr{A}^n, \boldsymbol{x}_{1,t-1}=\boldsymbol{y}_{1,t-1}} \mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}(\boldsymbol{x})}$$

Or, il est possible de factoriser les comptages issus du mot $y_{1,t-1}$ dans les deux membres de la fraction, puisqu'ils sont identiques pour toutes les séquences sur lesquelles la somme est effectuée :

$$\mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}(\{\boldsymbol{x}, \boldsymbol{x}_{1,t} = \boldsymbol{y}_{1,t}\}|\{\boldsymbol{x}, \boldsymbol{x}_{1,t-1} = \boldsymbol{y}_{1,t-1}\}) =$$

$$\frac{\sum\limits_{\boldsymbol{x}\in\mathscr{A}^n, \boldsymbol{x}_{1,t}=\boldsymbol{y}_{1,t}} \exp - \sum_{a\in\mathscr{A}} \mu_a N_a(\boldsymbol{x}_{t,n}) - \sum_{a_1,a_2\in\mathscr{A}^2} \mu_{a_1a_2} N_{a_1a_2}(\boldsymbol{x}_{t,n}) - \mu_{y_t} - \mu_{y_{t-1},y_t}}{\sum\limits_{\boldsymbol{x}\in\mathscr{A}^n, \boldsymbol{x}_{1,t-1}=\boldsymbol{y}_{1,t-1}} \exp - \sum_{a_1,a_2\in\mathscr{A}^2} \mu_{a_1a_2} N_{a_1a_2}(\boldsymbol{x}_{t,n}) - \mu_{x_t} - \mu_{y_{t-1}x_t}} \tag{13}$$

d'où :

$$\mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}(\{\boldsymbol{x}, \boldsymbol{x}_{1,t} = \boldsymbol{y}_{1,t}\}|\{\boldsymbol{x}, \boldsymbol{x}_{1,t-1} = \boldsymbol{y}_{1,t-1}\}) = \frac{\exp - \mu_{y_t} - \mu_{y_{t-1}y_t}}{\sum_{x_t\in\mathscr{A}} \exp - \mu_{x_t} - \mu_{y_{t-1}x_t}}$$

Comme cette dernière quantité ne dépend pas du début de la séquence $\boldsymbol{y}_{1,t-2}$, la distribution $\mathbb{P}^\star_{\langle \boldsymbol{N}\rangle}$ vérifie donc bien la propriété de MARKOV. □

2.4. Extension au cas $k > 2$. Nous n'avons jusqu'à présent envisagé que des contraintes liées à l'observation des comptages des mots de deux lettres. Il est cependant possible de traiter de manière parfaitement similaire le cas de comptages de mots de k lettres.

Les observations sont alors constituées du vecteur des comptages des mots de longueur 1 à k, $(N_{a_1}, N_{a_1 a_2}, \ldots, N_{a_1 \ldots a_k})_{(a_1,\ldots,a_k)\in\mathscr{A}^k}$. La distribution sur les séquences de longueur $n > k$ dont l'entropie est maximale s'écrit alors :

$$\forall \boldsymbol{x} \in \mathscr{A}^n, \mathbb{P}(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{\mu})} \exp - \sum_{i=1}^{k} \sum_{\boldsymbol{a}\in\mathscr{A}^i} \mu_{\boldsymbol{a}} N_{\boldsymbol{a}}(\boldsymbol{x})$$

Cette distribution vérifie alors la propriété de MARKOV *à l'ordre k*, ce qui signifie que :

$$\mathbb{P}(\{\boldsymbol{x}, \boldsymbol{x}_{1,t} = \boldsymbol{y}_{1,t}\} | \{\boldsymbol{x}, \boldsymbol{x}_{1,t-1} = \boldsymbol{y}_{1,t-1}\}) = \mathbb{P}(\{\boldsymbol{x}, \boldsymbol{x}_{1,t} = \boldsymbol{y}_{1,t}\} | \{\boldsymbol{x}, \boldsymbol{x}_{t-k+1,t-1} = \boldsymbol{y}_{t-k+1,t-1}\})$$

Cette propriété se vérifie immédiatement, en reprenant *mutatis mutandis* la démonstration précédente.

2.5. Résultats classiques. Compte-tenu que la *probabilité de transition* de x_{t-1} vers x_t s'écrit $\exp -\mu_{x_t} - \mu_{x_{t-1},x_t}$ à une constante près, et en particulier ne dépend pas de la position t ni de la longueur n de la séquence modélisée, on peut construire les chaînes de Markov en *chaînant* les probabilités de transition. C'est l'approche usuellement retenue pour construire les chaînes de MARKOV. Nous présentons ici l'approche probabiliste des chaînes de MARKOV, qui tient un rôle complémentaire au principe de maximum d'entropie. Ce dernier nous a en effet permis de choisir une distribution particulière pour décrire notre connaissance sur les séquences apportée par le comptage des mots de k lettres, mais il reste à s'assurer que sous cette distribution sur les séquences, cette même distribution serait retrouvée par la même approche.

DÉFINITION 4. *Une chaîne de Markov d'ordre k est une séquence $\boldsymbol{X} = (X_i)_{i\in\mathbb{N}^\star}$, telle qu'il existe une application $\pi : \mathscr{A}^k \times \mathscr{A} \to [0,1]$ et une mesure de probabilité $\mathbb{P}_0$ sur $\mathscr{A}^k$ pour lesquels les relations :*

$$\forall \boldsymbol{w} \in \mathscr{A}^k, \forall a \in \mathscr{A}, \forall t > k, \mathbb{P}(X_t = a | \boldsymbol{X}_{t-k,t-1} = \boldsymbol{w}) = \pi(\boldsymbol{w}, a) \tag{14}$$

et

$$\forall \boldsymbol{w} \in \mathscr{A}^k, \mathbb{P}(\boldsymbol{X}_{1,k} = \boldsymbol{w}) = \mathbb{P}_0(\boldsymbol{w}) \tag{15}$$

sont vérifiées.

La distribution $\mathbb{P}_0$ est appelée la distribution initiale de la séquence. L'application π est classiquement représentée comme une matrice de dimensions $|\mathscr{A}|^k \times |\mathscr{A}|$ appelée matrice de transition *de la chaîne de Markov. Pour chaque $\boldsymbol{w} \in \mathscr{A}^k$, l'application $\pi(\boldsymbol{w}, .) : \mathscr{A} \to [0,1]$, que l'on note également $\pi_{\boldsymbol{w}}$, définit une mesure sur l'alphabet. La matrice de transition est donc une matrice stochastique : les sommes de ses lignes sont toutes égales à 1.*

2.5.1. *Estimation.* Considérons à présent une trajectoire finie $\boldsymbol{x} \in \mathscr{A}^n$ d'une chaîne de Markov. Pour tout mot de k lettres $\boldsymbol{w} \in \mathscr{A}^k$, on note $\mathscr{I}(\boldsymbol{w})$ l'ensemble des indices des symboles de la trajectoire qui sont précédés du mot $\boldsymbol{w}$:

$$\mathscr{I}(\boldsymbol{w}, \boldsymbol{x}) = \{t | k < t < n \text{ et } \boldsymbol{x}_{t-k,t-1} = \boldsymbol{w}\}$$

L'ensemble $\mathscr{S}(\boldsymbol{w}, \boldsymbol{x})$ des symboles situés en ces positions de la séquence $\boldsymbol{x}$ forment alors un échantillon de la distribution $\pi_{\boldsymbol{w}}$. Une chaîne de Markov est donc un mélange de distributions sur l'alphabet $\mathscr{A}$, la composante de ce mélange étant déterminée en chaque position par les k lettres précédentes dans la séquence.

Du point de vue statistique, le modèle de Markov permet de probabiliser l'ensemble des séquences de longueur finie n. Etant donnée une séquence $\boldsymbol{x}$ de longueur n, la vraisemblance $L_{\boldsymbol{x}}(\mu,\boldsymbol{\pi})$ du couple de paramètres $(\mu,\boldsymbol{\pi})$ s'écrit :

$$L_{\boldsymbol{x}}(\mu,\boldsymbol{\pi}) = \mu(\boldsymbol{x}_{1,k}) \prod_{\boldsymbol{w}\in\mathscr{A}^k} \prod_{a\in\mathscr{A}} \pi_{\boldsymbol{w}}(a)^{N(\boldsymbol{w}a)} \tag{16}$$

où les quantités $N(\boldsymbol{w}a)$ désignent les comptages des mots $\boldsymbol{w}a$ dans la séquence $\boldsymbol{x}$. L'équation (16) est une forme factorisée de la vraisemblance indexée par les positions :

$$L_{\boldsymbol{x}}(\mu,\boldsymbol{\pi}) = \mu(\boldsymbol{x}_{1,k}) \prod_{t=k+1}^{n} \pi_{\boldsymbol{x}_{t-k,t-1}}(x_t)$$

On retrouve dans la formule (16) la remarque précédente : le produit sur les mots $\boldsymbol{w} \in \mathscr{A}^k$, que l'on appelle les *contextes* de la chaîne de Markov, traduit l'indépendance des échantillons conditionnels $\mathscr{S}(\boldsymbol{w},\boldsymbol{x})$. Pour chaque distribution conditionnelle $\pi_{\boldsymbol{w}}$, la factorisation montre que seul le comptage des occurences de chacune des lettres dans l'échantillon $\mathscr{S}(\boldsymbol{w},\boldsymbol{x})$ importe. Les comptages des mots de $k+1$ lettres dans la séquence constitue une statistique exhaustive et minimale du modèle de Markov.

La vraisemblance étant définie, l'estimation par le maximum de vraisemblance du paramètre de transition de la chaîne de Markov ne requiert qu'une maximisation. Nous allons voir que cette estimation coïncide précisément avec l'estimation de chacune des distributions conditionnelles aux contextes.

THÉORÈME 10. *L'estimateur du maximum de vraisemblance $\hat{\boldsymbol{\pi}}$ de la matrice de transition de la chaîne de Markov d'ordre k en fonction de la réalisation finie $\boldsymbol{x} = (x_1,\ldots,x_n)$ s'écrit :*

$$\hat{\pi}_{\boldsymbol{w}}(a) = \frac{N(\boldsymbol{w}a)}{\sum_{a\in\mathscr{A}} N(\boldsymbol{w}a)} \tag{17}$$

La preuve de ce théorème ne requiert que la maximisation de la log-vraisemblance sous la contrainte de stochasticité de la matrice de transition :

$$\begin{aligned} &\max \log L_{\boldsymbol{x}}(\mu,\boldsymbol{\pi}) \\ &s.c.\ \forall \boldsymbol{w}\in\mathscr{A}^k,\ \sum_{a\in\mathscr{A}} \pi_{\boldsymbol{w}}(a) = 1 \end{aligned} \tag{18}$$

Le lagrangien associé au problème d'optimisation (18), noté $\mathscr{L}$, s'écrit :

$$\mathscr{L}(\boldsymbol{\pi},\boldsymbol{\lambda}) = \log L_{\boldsymbol{x}}(\mu,\boldsymbol{\pi}) + \sum_{\boldsymbol{w}\in\mathscr{A}^k} \lambda_{\boldsymbol{w}} \left(\sum_{a\in\mathscr{A}} \pi_{\boldsymbol{w}}(a) - 1 \right)$$

où $\boldsymbol{\lambda} = (\lambda_{\boldsymbol{w}})_{\boldsymbol{w}\in\mathscr{A}^k}$ est le multiplicateur de Lagrange associé à la contrainte de stochasticité de la matrice $\boldsymbol{\pi}$.

Le système d'optimalité associé s'écrit alors :

$$\frac{\partial \mathscr{L}}{\partial \pi_{\boldsymbol{w}}(a)} = \frac{N(\boldsymbol{w}a)}{\hat{\pi}_{\boldsymbol{w}}(a)} - \lambda_{\boldsymbol{w}} = 0, \quad \boldsymbol{w}\in\mathscr{A}^k, a\in\mathscr{A} \tag{19}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda_{\boldsymbol{w}}} = \sum_{a\in\mathscr{A}} \hat{\pi}_{\boldsymbol{w}}(a) - 1 = 0, \quad \boldsymbol{w}\in\mathscr{A}^k \tag{20}$$

De l'équation (19) on tire la relation :

$$\forall \boldsymbol{w}\in\mathscr{A}^k, \forall a\in\mathscr{A}, \hat{\pi}_{\boldsymbol{w}}(a) = \frac{N(\boldsymbol{w}a)}{\lambda_{\boldsymbol{w}}}$$

et l'équation (20) implique alors que :

$$\forall \boldsymbol{w}\in\mathscr{A}^k, \lambda_{\boldsymbol{w}} = \sum_{a\in\mathscr{A}} N(\boldsymbol{w}a)$$

d'où le résultat. □

Bien entendu, la relation (17) ne vaut que si tous les contextes $\boldsymbol{w} \in \mathscr{A}^k$ apparaissent effectivement dans la séquence $\boldsymbol{x}$: on obtient sinon un quotient indéterminé. Dans un tel cas, la maximisation de la vraisemblance n'impose pas de solution. Les approches bayésiennes que nous présenterons plus loin fourniront une solution générale à ce type d'indétermination. Cependant le principe de maximum d'entropie dicte déjà de considérer, dans ce cas, que la distribution est uniforme.

2.5.2. *Ergodicité.* Nous avons vu que le principe de maximum d'entropie conduisait, dans le cas de séquences observées par le prisme des comptages des mots de k lettres, à envisager le modèle de MARKOV d'ordre k. Ce faisant, on peut s'interroger sur le risque de surestimation de l'entropie : ne peut-il y avoir plus de régularités dans le processus que celles identifiées par cette approche ? Bien entendu, il ne s'agit pas de trancher cette question pour chaque séquence explicite étudiée. En revanche, si, sous l'hypothèse qu'elle est générée selon une chaîne de MARKOV, une séquence aléatoire ne *sature* pas la diversité prédite par ce modèle, on peut questionner la cohérence de la démarche. Formellement, cette idée se traduit par la propriété d'ergodicité des distributions obtenues par maximum d'entropie. L'ergodicité traduit en effet l'adéquation asymptotique entre les fréquences d'occurrence des états et leur probabilité.

Ce type de résultat s'obtient en étudiant la trajectoire (en fonction de la position t) de la distribution marginale $\mu_t(\boldsymbol{a})$ de la séquence, pour $a \in \mathscr{A}$ et $t \in \mathbb{N}^\star$. La dynamique de cette mesure résulte simplement de l'action de la matrice de transition comme le montre le théorème suivant.

THÉORÈME 11. *Soit* $(X_1, X_2, \ldots)$ *une chaîne de Markov sur l'alphabet* $\mathscr{A}$ *de matrice de transition* $\boldsymbol{\pi}$. *La mesure marginale* $\mathbb{P}_t = (\mathbb{P}_t(\boldsymbol{w}))_{\boldsymbol{w} \in \mathscr{A}^k}$ *définie par la relation :*

$$\forall \boldsymbol{w} \in \mathscr{A}^k, \forall t > k, \mathbb{P}_t(\boldsymbol{w}) = \mathbb{P}(\boldsymbol{X}_{t-k+1,t} = \boldsymbol{w})$$

répond à la dynamique :

$$\mathbb{P}_t = \mathbb{P}_{t-1}\boldsymbol{\Pi} \tag{21}$$

où l'on a noté Π *la matrice de transition* étendue, *de dimension* $|\mathscr{A}|^k \times |\mathscr{A}|^k$, *définie par la relation* $\Pi(\boldsymbol{w}, \boldsymbol{w}') = \mathbb{P}(\boldsymbol{X}_{t-k+1,t} = \boldsymbol{w}' | \boldsymbol{X}_{t-k,t-1} = \boldsymbol{w})$.

La preuve de ce théorème tient à un simple calcul. En effet pour tout $\boldsymbol{w}' \in \mathscr{A}^k$:

$$\forall t > k, \mathbb{P}(\boldsymbol{X}_{t-k+1,t} = \boldsymbol{w}') = \sum_{w'_0 \in \mathscr{A}} \mathbb{P}(\boldsymbol{X}_{t-k,t} = (w'_0, \ldots, w'_k))$$

Or

$$\forall w_1 \in \mathscr{A}, \mathbb{P}(\boldsymbol{X}_{t-k,t} = (w_1, \ldots, w'_k) = \mathbb{P}(\boldsymbol{X}_{t-k,t-1} = (w_1, \ldots, w'_{k-1})) \times \pi_{(w_1,\ldots,w'_{k-1})}(w'_k)$$

d'où :

$$\mathbb{P}(\boldsymbol{X}_{t-k+1,t} = \boldsymbol{w}') = \sum_{w_1 \in \mathscr{A}} \mathbb{P}(\boldsymbol{X}_{t-k,t-1} = (w_1, w'_1, \ldots, w'_{k-1})) \times \pi_{(w_1,w'_1,\ldots,w'_{k-1})}(w'_k) \tag{22}$$

Comme pour tout $\boldsymbol{w} \in \mathscr{A}^k$, on a :

$$\mathbb{P}(\boldsymbol{X}_{t-k+1,t} = \boldsymbol{w}' | \boldsymbol{X}_{t-k,t-1} = \boldsymbol{w}) = \mathbf{1}_{(w_2,\ldots,w_k)=(w'_1,\ldots,w'_{k-1})} \pi_{\boldsymbol{w}}(w'_k)$$

on peut réécrire l'équation (22) en séparant les mots $\boldsymbol{w} = (w_1, w'_1, \ldots, w'_{k-1})$ et $\boldsymbol{w}'$, ce qui conduit à la relation :

$$\mathbb{P}(\boldsymbol{X}_{t-k+1,t} = \boldsymbol{w}') = \sum_{\boldsymbol{w} \in \mathscr{A}} \mathbb{P}(\boldsymbol{X}_{t-k,t-1} = \boldsymbol{w}) \times \Pi(\boldsymbol{w}, \boldsymbol{w}')$$

soit encore $\mathbb{P}_t = \mathbb{P}_{t-1}\Pi$. □

Si l'on initialise cette dynamique avec la mesure initiale sur les mots de k-lettres $\mathbb{P}_0$, on obtient l'équation suivante :

$$\forall t > k, \mathbb{P}_t = \mathbb{P}_0 \Pi^{t-k}$$

En particulier, partant d'une position t où se termine le mot $\boldsymbol{w}$, le coefficient $(\boldsymbol{w},\boldsymbol{w}')$ de la matrice $\Pi^{t'-t}$ est égal à la probabilité qu'une occurence du mot $\boldsymbol{w}'$ se termine en position t'.

Dans le cas particulier où, pour une puissance l, tous les coefficients de la matrice Π^l sont strictement positifs, il est alors manifeste que l'ensemble des mots ont une probabilité non-nulle d'apparaître dans un segment de longueur l de la séquence. Et ce quel que soit le mot initiant ce segment, et quelle que soit la position du segment dans la séquence. On retrouve là une manifestation de la propriété de Markov. Compte-tenu du fait que la matrice $\mathbf{\Pi}$ est stochastique, tout comme sa version compacte $\boldsymbol{\pi}$, cette propriété reste vraie pour les puissances supérieures à l. C'est ce que montre la propriété suivante.

PROPOSITION 1. *Si la matrice stochastique $\mathbf{\Pi}$ admet une puissance $l \geq 1$ telle que $\mathbf{\Pi}^l$ n'a que des coefficients strictement positifs, alors les puissances suivantes de la matrice présentent la même propriété :*

$$\forall l' \geq l, \forall \boldsymbol{w},\boldsymbol{w}' \in \mathscr{A}^k, \Pi^{l'}(\boldsymbol{w},\boldsymbol{w}') > 0 \tag{23}$$

Cette propriété résulte d'une simple récurrence. La propriété (23) est vraie pour $l' = l$. Supposons là vraie pour $l' \geq l$. On a alors $\mathbf{\Pi}^{l'+1} = \mathbf{\Pi} \times \mathbf{\Pi}^{l'}$. Par conséquent :

$$\forall \boldsymbol{w},\boldsymbol{w}' \in \mathscr{A}^k, \Pi^{l'+1}(\boldsymbol{w},\boldsymbol{w}') = \sum_{\boldsymbol{w}'' \in \mathscr{A}^k} \Pi(\boldsymbol{w},\boldsymbol{w}'')\Pi^{l'}(\boldsymbol{w}'',\boldsymbol{w}')$$

Chaque coefficient $\Pi^{l'+1}(\boldsymbol{w},\boldsymbol{w}')$ est donc le barycentre des termes de la colonne $\boldsymbol{w}'$ de $\mathbf{\Pi}^{l'}$, qui sont par hypothèse tous strictement positifs. Le barycentre l'est donc aussi, ce qui permet de conclure la récurrence. □

Lorsque la propriété précédente est remplie, on dit que la matrice de transition de la chaîne de MARKOV est *primitive*. Mais plus généralement, la condition suivante suffit à assurer les principaux résultats de convergence.

DÉFINITION 5. *Une chaîne de* MARKOV $(X_1,\ldots,X_n)$ *sur un espace d'états fini $\mathscr{A}$ est dite irréductible si :*

$$\forall (i,j) \in \mathscr{A}^2, \exists n \in \mathbb{N}, (\pi^n)_{i,j} > 0 \tag{24}$$

Cette définition traduit simplement que tous les états de la chaîne communiquent avec une probabilité non-nulle, éventuellement moyennant plusieurs transitions successives. Cela a en particulier comme conséquence que, quelque soit l'initialisation de la chaîne, il est presque sûr qu'aucun état ne sera évité le long de celle-ci.

Sous l'hypothèse d'irréductibilité (ou a fortiori, celle de primalité), on peut appliquer à l'opérateur de transition π le théorème de PERRON-FROBENIUS, exposé ici dans le cas d'une matrice stochastique, donc de norme 1.

THÉORÈME 12. *Soit π une matrice carrée de taille $|\mathscr{A}|$. On suppose que π est irréductible, et que sa norme d'opérateur est 1. Alors π admet 1 pour valeur propre simple. Si π est de plus primale, alors toutes les autres valeurs propres de π sont de module strictement inférieur à 1.*

La preuve de ce théorème peut être trouvée dans [**?**].

La distribution $\mathbb{P}^\star$ sur les mots de k lettres qui est vecteur propre de π pour cette valeur propre est donc telle que :

$$\mathbb{P}^\star = \mathbb{P}^\star \pi$$

et constitue la *distribution stationnaire* de la chaîne de MARKOV de matrice de transition π. Le terme stationnaire renvoie au fait que, si la chaîne est initialisée avec la distribution $\mathbb{P}_0 = \mathbb{P}^\star$, alors :

$$\forall t > k, \mathbb{P}_t = \mathbb{P}^\star \pi^{t-k} = \mathbb{P}^\star$$

En fait, même lorsque la chaîne de MARKOV n'est pas initialisée avec la distribution stationnaire, on peut montrer, sous l'hypothèse qu'elle est primale, que la distribution marginale $\mathbb{P}_t$ converge vers cette même distribution stationnaire.

THÉORÈME 13. *Soit π la matrice de transition d'une chaîne de* MARKOV *primale. Alors, quelque soit la distribution initiale $\mathbb{P}_0$, on a :*

$$\mathbb{P}_t \longrightarrow_{t\to\infty} \mathbb{P}^\star$$

La convergence a lieu à vitesse exponentielle.

La preuve de ce théorème s'appuie fondamentalement sur le résultat du théorème de PERRON-FROBENIUS, et en particulier sur l'assertion que la valeur propre de plus grand module autre que 1 a un module strictement inférieur à 1 : la vitesse de convergence est en effet majorée par λ^t, où $\lambda < 1$ désigne ce module de la seconde valeur propre. Une preuve élégante de ce résultat peut être trouvée dans [**?**].

Cette situation correspond à l'ergodicité de la chaîne de MARKOV, puisqu'elle assure la convergence des moyennes empiriques de fonction de l'état de la chaîne vers leur espérance sous le modèle. Cette propriété est cruciale pour la convergence des estimateurs.

2.5.3. *Consistance du maximum de vraisemblance.*

THÉORÈME 14. *L'estimateur du maximum de vraisemblance est consistant pour l'estimation d'une chaîne de Markov ergodique. Formellement, si $\boldsymbol{\pi}$ désigne une matrice de transition, et $\mathbb{P}$ désigne la distribution induite sur les séquences de longueur n sur l'alphabet $\mathscr{A}$, alors :*

$$\hat{\boldsymbol{\pi}} \rightarrow_{p.s.} \pi$$

dès lors que $\mathbb{P}$ est ergodique.

Une preuve de ce théorème fondée sur la méthode des types peut être trouvée dans [**?**].

CHAPITRE 3

L'analyse statistique des séquences biologiques

Après cette rapide description des caractéristiques les plus saillantes ayant été identifiées dans la composition des génomes et de leurs liens avec les propriétés biologiques codées sur le génome, nous présentons à présent les méthodes de prédiction de ces propriétés qui en ont été dérivées. Dans un premier temps, nous aborderons les méthodes utilisées pour la détection des gènes dans les génomes, en prolongement des approches introduites par SHEPHERD et évoquées précédemment. Nous aborderons ensuite la prédiction de la fonction des gènes par comparaison de séquences et les questions statistiques qui lui sont associées. Enfin, nous présenterons les méthodes utilisées pour l'analyse des séquences protéiques, et plus particulièrement celles permettant de réaliser une prédiction *ab initio* de la structure secondaire de la protéine.

1. Détection de gènes par classification

Compte-tenu de la vitesse extrêmement rapide d'acquisition des nouvelles séquences génomiques complètes, il est nécessaire de disposer des meilleures méthodes de détection automatique des gènes. A la suite des travaux de STADEN et SHEPHERD, le sujet a été amplement exploré, et aujourd'hui les outils issus de la bioinformatique pour ce faire sont assez aboutis : les détecteurs de gène utilisés au quotidien présentent une sensibilité de l'ordre de 95%, pour une spécificité totale ou quasiment totale (on compte au pire quelques faux positifs). Le défaut de sensibilité tient principalement à la présence de petits gènes dont le signal statistique est trop faible pour est considéré comme significatif.

1.1. Un problème de classification. Compte-tenu de la *ponctuation* décrite précédemment, et qui permet la définition des cadres ouverts de lecture, déterminer l'ensemble des séquences codantes d'un génome se réduit à un problème de classification : parmi les cadres ouverts de lecture, il s'agit de reconnaître ceux qui sont effectivement des séquences codantes, et ceux qui n'en sont pas. Deux types de signaux peuvent supporter l'hypothèse qu'un cadre ouvert de lecture est une région codante : ceux portés par la séquence elle-même, et associés aux biais de composition intrinsèques aux séquences codantes, et d'autre part ceux portés par le voisinage de la région codante, permettant la fixation de la machinerie d'expression de l'information génétique.

1.1.1. *Modèles.* Nous nous intéressons dans un premier temps aux modèles qui permettent de capturer le premier phénomène. Nous avons vu au chapitre précédent que le modèle adéquat pour prendre en compte les biais de compositions de la séquence en mots de quelques lettes était les chaînes de Markov. Aussi, une méthode élémentaire mais efficace pour réaliser cette classification consiste à ajuster un modèle de Markov pour les régions codantes, un modèle de Markov pour les régions non-codantes, et de réaliser un test de rapport de vraisemblance pour affecter un cadre ouvert de lecture à l'une des deux classes, codant ou non-codant.

Le principal obstacle pour la mise en œuvre d'une telle méthode est l'apprentissage des matrices de transition spécifiques de chaque classe, puisque l'on ne sait pas a priori quelles sont les régions codantes. Cependant, l'approximation supérieure de l'ensemble des régions codantes formée par l'ensemble des cadres ouverts de lecture

est en grande majorité formée de régions effectivement codantes (rappelons que près de 95% du génome d'une bactérie est codante). Cette approximation supérieure peut être réduite de deux manières :

- en se restreignant aux cadres ouverts de lecture dont la longueur est significativement longue au seuil de 5% sous un modèle d'indépendance des bases. Selon la composition de la séquence, cela conduit à des longueurs minimales de l'ordre de quelques centaines de nucléotides,
- en se restreignant aux cadres ouverts de lecture qui n'en contiennent pas d'autre, afin d'éviter que les cadres ouverts contenant un plus grand nombre de codons `START` en phase ne soient artificiellement surreprésentés. On parle alors de cadres ouverts de lecture *maximaux*.

1.1.2. *Une méthode de classification des cadres ouverts de lecture.* Ainsi, une procédure élémentaire permet de réaliser une détection des gènes assez fidèle. Connue sous le nom de PROKOV, elle est due à A .VIARI, et, si elle n'a pas été publiée, elle est néanmoins au cœur de la plate-forme d'annotation AMIGENE ([**?**]). Elle procède comme suit :

- détecter les cadres ouverts de lecture dont la longueur excède le seuil de 5% sous un modèle d'indépendance des nucléotides le long de la séquence,
- estimer un modèle de Markov $\mathbb{P}_c$ phasé (avec une matrice de transition différente pour chaque phase de lecture) à partir de cet ensemble de cadres ouverts,
- estimer un modèle de Markov $\mathbb{P}_{nc}$ pour les régions non-codantes, en utilisant pour cela le génome privé des plus longues ORF.
- pour chaque ORF dont la longueur est suffisante, calculer ses probabilités $\mathbb{P}_c(\boldsymbol{x})$ et $\mathbb{P}_{nc}(\boldsymbol{x}$ sous les modèles de séquence codante et de séquence non-codante respectivement,
- calculer le rapport de vraisemblance $R(\boldsymbol{x}) = \log\mathbb{P}_c(\boldsymbol{x}) - \log\mathbb{P}_{nc}(\boldsymbol{x})$, et classer le cadre ouvert comme codant si $R(\boldsymbol{x}) > s$, et comme non-codant sinon.

1.1.3. *Chevauchement des séquences codantes.* Nous avons vu précédemment que les cadres ouverts de lecture pouvaient être emboîtés les uns dans les autres : en effet, dès lors qu'un cadre ouvert de lecture contient un second codon d'initiation de la traduction en phase, il contient un second cadre ouvert de lecture commençant en ce second codon d'initiation de la traduction et terminant au même codon de terminaison. De manière plus générale, plusieurs cadres ouverts de lecture peuvent partager le même codon de terminaison. A l'issue la classification des cadres ouverts de lecture, il est donc possible que plusieurs cadre de lecture emboités présentent un rapport de vraisemblance supérieur au seuil de significativité. Dans le même ordre d'idée, des cadres de lecture peuvent sembler *significativement codants* à cause des contraintes que leur impose la présence d'une séquence codante sur une autre phase de lecture. Ceci conduit à des situations telles que celle représentée sur la figure 1.A, sur laquelle apparaissent clairement des *séquences vraisemblablement codantes* chevauchantes.

Or, du point de vue biologique, ce type de situation est extrêmement rare : il pourrait en effet conduire à une collision d'ARN-polymérases transcrivant un gène sur chaque brin simultanément. AMIGene procède donc à une élimination automatique des CDS fantômes de la manière suivante. Les séquences vraisemblablement codantes sont, pour ce faire, divisées en deux groupes : celles dont le rapport de vraisemblance est très en faveur du modèle codant, caractérisées par le fait que ce dernier dépasse un seuil s_{sure} ; et celles dont le rapport de vraisemblance est inférieur à ce seuil, mais cependant vraisemblablement codantes (nommées CDS *probables* dans la suite). La démarche est alors la suivante :

- Les CDS sures sont comparées en terme de position. Lorsqu'un chevauchement entre CDS de sens opposé est identifié, et que l'une des CDS contient l'autre, la plus grande est conservée.

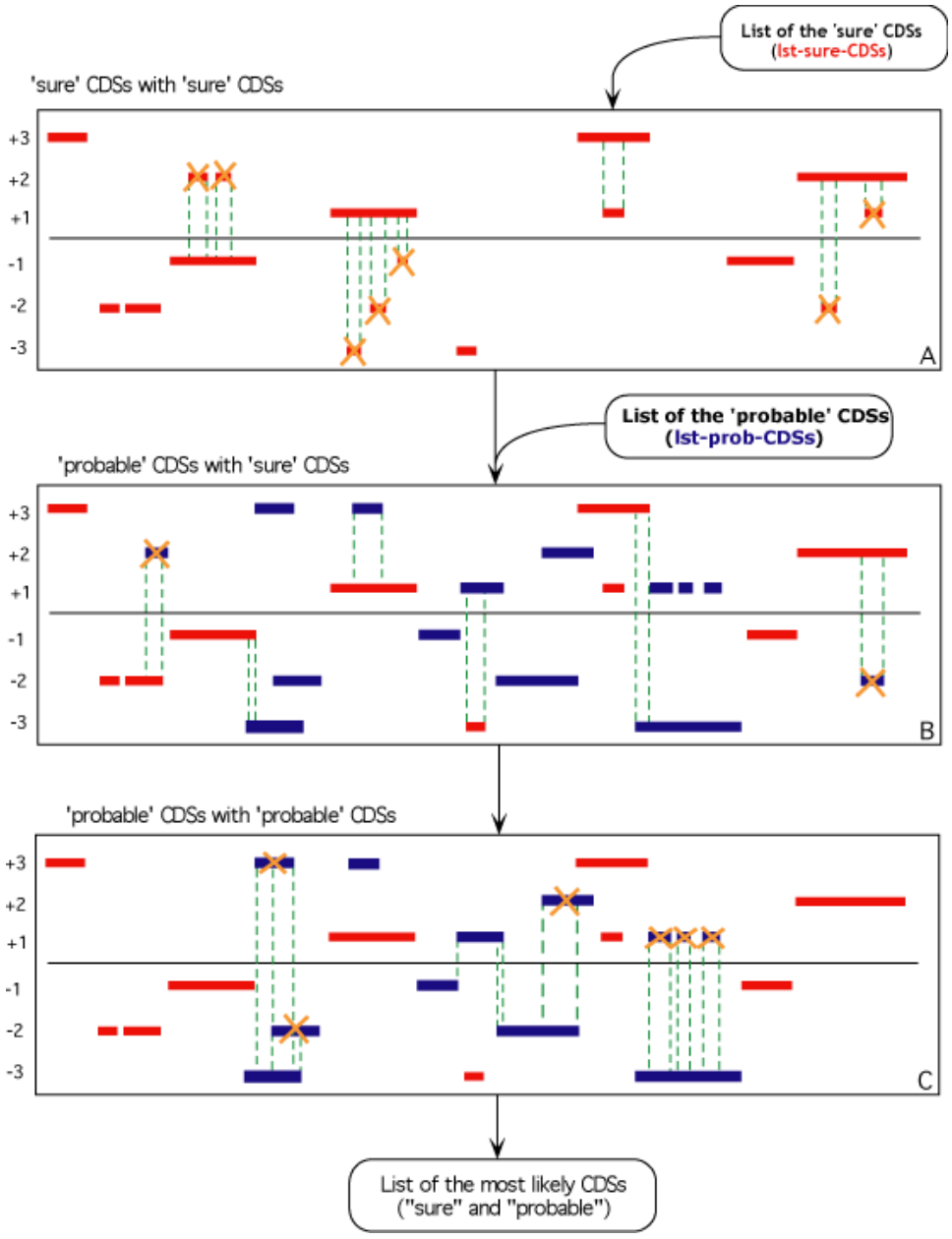


FIG. 1. La procédure d'élimination des séquences codantes artefactuelles dues au chevauchement des cadres ouverts de lecture

– Les CDS sures sont ensuite comparées aux CDS probables. La même procédure d'élimination est appliquée, sauf que l'on élimine toujours la CDS probables au profit de la CDS sure.
– Les CDS probables sont enfin comparées entre elles, et la même procédure d'élimination est appliquée que pour les CDS sures.

L'ensemble de la procédure AMIGene produit ainsi un ensemble cohérent de CDS qui présentent les propriétés suivantes :

– elles sont comprises entre un codon START et un codon STOP séparés d'un nombre de nucléotides multiple de 3, sans qu'un codon STOP n'apparaissent en phase avec ces deux codons à l'intérieur de la CDS,
– elles présentent un profil d'usage du code génétique proche de l'usage moyen du code sur l'ensemble du génome,
– elles ne se chevauchent pas entre brins opposés.

1.2. Limites des approches par classification des cadres ouverts de lecture.

1.2.1. *Petits gènes.* Usuellement, la détection des cadres ouverts de lecture s'accompagne d'un filtrage selon leur longueur : plus long est un cadre ouvert de lecture, et plus faible est la probabilité de l'évitement du codon STOP à l'intérieur de la phase de lecture sous un modèle d'indépendance des nucléotides de la séquence. De ce fait, on requiert usuellement une longueur minimale de quelques centaines de bases (de l'ordre de 300) pour soumettre un cadre ouvert de lecture à l'évaluation de la probabilité qu'il s'agisse d'une séquence codante. Cependant, certains organismes présentent quelques gènes d'une longueur voisine des seuils usuellement utilisés.

Qui plus est, ces *petits gènes* peuvent, du fait de leur taille, présenter un biais d'usage des codons moins significatif que pour des gènes plus longs, ce qui rend leur classification en codant/non-codant plus délicate. Les petits gènes constituent, pour toutes ces raisons, des régions plus difficiles à détecter.

1.2.2. *Correction du codon* START. Lorsque plusieurs signaux *START* peuvent initier un gène, les relations d'inclusion entre les différents cadres ouverts de lecture résultant du choix du codon d'initiation peuvent conduire à considérer comme *surement codant* plusieurs cadre de lecture "concurrents", c'est-à-dire partageant le même codon STOP. Les règles d'élimination présentées ci-dessus peuvent alors conduire à retenir un codon START trop loin en amont, puisque la séquence codante la plus longue est retenue.

Même si ce type de situation peut être décelé en aval, en particulier lors de l'annotation fonctionnelle des gènes qui peut révéler que l'ensemble des homologues connus du gène ne partagent pas le début de la séquence avec le gène annoté et ainsi conduire à réannoter le START du gène, il reste utile de réaliser la meilleure prédiction possible de ce codon d'initiation. Nous allons voir à présent que divers aspects moléculaires de l'expression des gènes fournissent des éléments pour ce faire.

1.2.3. *Signaux environnant les gènes.* L'ARN-polymerase, le complexe protéique responsable de la transcription des gènes en ARN, présente une affinité structurale pour l'ADN. Cependant, cette affinité n'est pas spécifique de sites particuliers de la molécule d'ADN. En revanche, l'ARN-polymerase s'assemble avec un facteur appelé *sigma,* qui, lui, présente une affinité spécifique pour des régions dites *promotrices* du gène et situées en amont de celui-ci. Les séquences promotrices ont fait l'objet de nombreuses investigations, qui ont permis d'identifier des *consensus.* En particulier, deux séquences consensus sont fréquemment mentionnées :

– la séquence -35 : présente typiquement 35 bases en amont du site d'initiation de la traduction, elle est composée des nucléotides TTGACA.
– la séquence -10 (ou TATA-box) : présente 10 bases en amont du site d'initiation de la traduction, elle est formée par les nucléotides TATAAT.

Il est important de remarquer que ces séquences sont des consensus seulement, ce qui siginifie que de faibles variations de ces séquences sont régulièrement observées. Par ailleurs, selon l'importance du gène et selon la nécessité de l'exprimer de manière permanente ou transitoire, ces séquences peuvent être plus ou moins dégénérées, voire partiellement absentes. Ceci est en particulier vrai pour la séquence -35, pour laquelle d'autre formes ont été identifiées également.

De même, l'affinité du ribosome avec le brin d'ARN est conditionnée par une séquence particulière, appelée *séquence consensus de* SHINE-DALGARNO chez les procaryotes. Elle est fomée par le motif AGGAGG, située 7 bases avant le codon START du gène.

Ces signaux, qui flanquent de manière plus ou moins obligatoire les gènes d'un organisme, sont autant d'indice qui peuvent en particulier permettre de préciser la position du codon START d'un gène, voire valider ou invalider le caractère codant d'un cadre ouvert de lecture. Mais ils peuvent également apporter du support à la détection des pseudo-gènes, que nous introduisons à présent.

1.2.4. *Pseudo-gènes.* Au cours de l'évolution, les organismes acquièrent, mais aussi perdent des gènes : lorsque les descendants d'un organisme colonisent une nouvelle niche écologique de laquelle un nutriment précédemment usuel est absent, plus aucune pression sélective ne s'exerce sur les gènes responsables de la métabolisation de ce composé chimique : ce gène peut muter, cela n'affectera pas la viabilité de l'organisme. Compte-tenu de la vitesse d'évolution des séquences, des *traces* de ce gène persistent dans le génome des descendants. Ces traces sont typiquement affectées de mutation, mais aussi de délétion ou d'insertions. Une conséquence typique de ces délétions et insertions est d'introduire un décalage de la phase de lecture, c'est-à-dire que les codons START et STOP ne sont plus en phase. Si les mécanismes d'expression des gènes dans les organismes eucaryotes permettent d'exploiter ces *décalages de phase de lecture*, il n'en est pas de même chez les bactéries. Dans ce dernier cas, on parle de pseudogènes car, s'ils présentent encore de nombreuses caractéristiques des séquences codantes, le décalage de la phase de lecture empêche toute expression du gène.

Selon les objectifs de l'annotation, l'identification des pseudo-gènes ne revêt pas le même intérêt : lorsque l'objectif est de comprendre les compétences métaboliques de l'organisme tel qu'il est aujourd'hui, les pseudo-gènes ne sont pas recherchés. Ne pouvant être exprimés, ils ne peuvent en effet contribuer à ces capacités. En revanche, lorsque la séquence du génome d'un organisme est étudiée afin de mieux comprendre l'histoire évolutive de l'organisme, leur identification est un élément fondamental.

2. Détection de gènes par chaînes de Markov cachées

Prendre en compte l'ensemble de ces signaux, qui ne présentent pas tous la même variabilité, n'est pas trivial dans l'approche mise en œuvre par AMIGene. Une alternative à cette approche s'est développée depuis le début des années 1990, qui s'appuie sur une complexification du modèle de chaînes de Markov utilisé jusqu'ici pour la classification des cadres ouverts de lecture.

2.1. Définition. Les chaînes de Markov permettent de modéliser la succession de distributions différentes le long d'une séquence. La succession de ces régimes est alors définie grâce à une nouvelle séquence, appelée *séquence cachée*. Cette désignation traduit l'idée que la succession de ces régimes est inconnue du modélisateur, qui n'a accès qu'à la séquence dite *observée*. Formellement, cette extension répond à la définition suivante.

DÉFINITION 6. *Soit $J = \{1,\dots,p\}$, $\mathscr{A}$ un alphabet fini, $\pi_1,\dots,\pi_p$ p distributions sur $\mathscr{A}$, et π_S une matrice stochastique de taille $|J| \times |J|$. Les suites de variables aléatoires $\boldsymbol{X} = (X_t)_{1\le t\le n}$ et $\boldsymbol{S} = (S_t)_{1\le t\le n}$ à valeur dans $\mathscr{A}^n$ et J^n respectivement forment une* chaîne de Markov cachée *à p états cachés si :*

$$\forall t \in \{1,\dots,n\}, \forall \boldsymbol{x}_{1,t} \in \mathscr{A}^t, \forall \boldsymbol{s}_{1,t} \in \{1,\dots,p\}^t, \mathbb{P}(X_t = x_t | \boldsymbol{X}_{1,t-1} = x_{1,t-1}, \boldsymbol{S}_{1,t} = s_{1,t}) = \mathbb{P}(X_t = x_t | S_t = s_t) = \pi_{S_t}(x_t)$$

et si

$$\forall t \in \{1,\dots,n\}, \forall \boldsymbol{s}_{1,t} \in J^t, \mathbb{P}(S_t = s_t | \boldsymbol{S}_{1,t-1} = \boldsymbol{s}_{1,t-1}) = \mathbb{P}(S_t = s_t | S_{t-1} = s_{t-1})$$

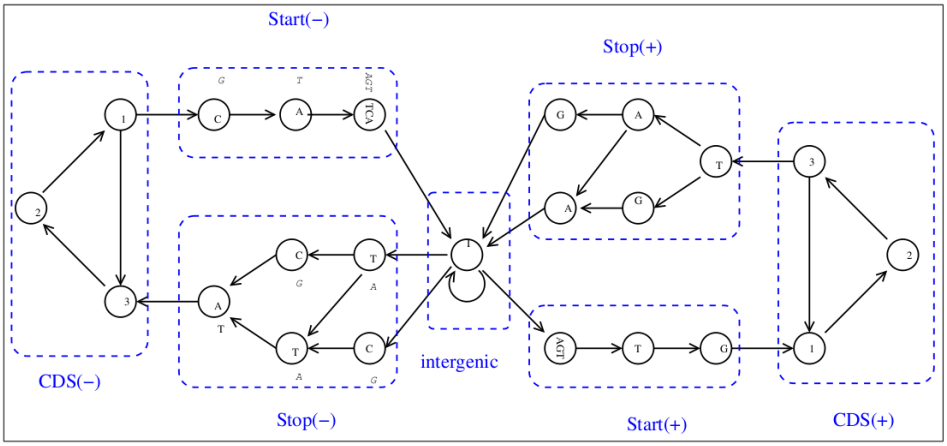


FIG. 2. Exemple de spécification des transitions entre états cachés pour une chaîne de Markov cachée destinée à la détection des gènes. Chaque noeud du graphe représente un état caché, et une flèche entre deux états indiquent que la probabilité de transiter de l'état à l'origine de la flèche vers l'état cible de la flèche est non nulle.

Ainsi, la valeur de la séquence $\boldsymbol{S}$ en position t dicte la distribution de la même position de la séquence observable, X_t. Une chaîne de Markov cachée à p états est par conséquent paramétrée par $p+1$ distributions : p distributions d'émission sur $\mathscr{A}$, et la distribution régissant les transitions entre états cachés. Mais les distributions d'émission, $\pi_1,\dots,\pi_p$, ne sont pas nécessairement sans mémoire. On peut en particulier construire des chaînes de Markov cachées dont les distributions d'émission sont elles-mêmes des chaînes de Markov d'ordre k comme suit.

DÉFINITION 7. *Soit $J=\{1,\dots,p\}$, $\mathscr{A}$ un alphabet fini, $\pi_1,\dots,\pi_p$ p matrices stochastiques de taille $|\mathscr{A}|\times|\mathscr{A}|$, et π_S une matrice stochastique de taille $|J|\times|J|$. Les suites de variables aléatoires $X_1,\dots,X_n$ et $S_1,\dots,S_n$ à valeurs dans un alphabet fini $\mathscr{A}$ et dans J respectivement forment une* chaîne de Markov cachée *d'ordre k à p états cachés si :*

$$\forall t\in\{1,\dots,n\},\forall \boldsymbol{x}_{1,n}\in\mathscr{A}^n,\forall \boldsymbol{s}_{1,n}\in J^n,\mathbb{P}(X_t=a|\boldsymbol{X}_{1,t-1}=\boldsymbol{x}_{1,t-1})=\mathbb{P}(X_t=a|\boldsymbol{X}_{t-k,t-1}=\boldsymbol{x}_{t-k,t-1},S_t=s_t)=\pi_{s_t}(\boldsymbol{x}_{t-k,t-1}$$

et si :

$$\forall t\in\{1,\dots,n\},\forall \boldsymbol{s}_{1,n}\in\mathscr{J}^n,\mathbb{P}(S_t=s_t|\boldsymbol{S}_{1,t-1}=\boldsymbol{s}_{1,t-1})=\mathbb{P}(S_t=s_t|S_{t-1}=s_{t-1})=\pi_S(s_{t-1},s_t)$$

Remarquons qu'il n'y a aucune limitation à considérer le cas plus général de p distributions d'émission d'ordres markoviens respectifs $k_1,\dots,k_p$.

Dans ce cas, la distribution de la position t de la séquence observée $\boldsymbol{X}$ dépend des k_{s_t} lettres la précédent dans la séquence, et ce d'une manière dépendante de la valeur de la séquence cachée en cette position, S_t. Dans l'optique de réaliser une détection de gène, ce modèle peut se substituer à la classification des cadres ouverts de lecture. En considèrant trois états cachés le long de la séquence (non-codant, codant dans le sens direct, et codant dans le sens indirect), on permet en effet la succession de région où s'exerce une préférence du code avec des régions qui ne s'y conforment pas (voir figure 2).

2.2. Segmentation, décodage et estimation. Ces modèles ne seraient d'aucune utilité pratique sans la batterie de méthodes qui permettent leur exploitation efficace. Ces méthodes rendent possibles trois principaux types d'analyses :

– la *segmentation*, qui consiste à reconstituer le chemin caché sachant les paramètres de la chaîne de Markov cachée et la séquence observée,

– le *décodage a posteriori*, qui s'attache au calcul de la distribution des états cachés sachant les paramètres de la chaîne de Markov cachée et la séquence observée,
– et enfin l'estimation des paramètres sur *données incomplètes*, qui permettent l'estimation des $p+1$ distributions impliquées dans le modèle à p états cachés à partir de la seule connaissance de la séquence observée.

Chacun de ces types d'analyse répond à une préoccupation de la détection de gènes. En effet, dans un modèle à trois régimes `non-codant`, `codant +` et `codant -`, être capable de reconstruire le chemin caché signifie précisément détecter les gènes.

Cependant, la sélection du seul chemin caché le plus probable sachant la séquence observée peut conduire à des artefacts. En effet, le chemin caché le plus probable peut désigner une région comme non-codante, alors même que la somme des probabilités des chemins cachés qui la désignent codantes dépasse celle des alternatives. Cet exemple illustre qu'il y a plus d'information dans la distribution des chemins cachés sachant la séquence observée que dans la seule connaissance de son mode.

Enfin, les deux premiers types d'analyse requièrent la connaissance des paramètres de la chaîne de Markov cachée. Or, en général, le biais d'usage des codons qui caractérise les séquences codantes est dépendant de l'organisme, rendant délicate la transposition de paramètres appris sur un génome connu. Aussi, disposer d'un cadre d'estimation *non-supervisé* (c'est-à-dire, en l'occurrence, ne nécessitant pas de données segmentées pour estimer les paramètres) pour les chaînes de Markov cachées constitue un atout majeur.

Mais détaillons les méthodes permettant de réaliser ces analyses.

2.2.1. *Algorithme de* VITERBI. Nous présentons d'abord la principale méthode de segmentation, l'algorithme de VITERBI. A partir de la connaissance de la séquence observée, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathscr{A}^n$, et des paramètres $\pi_1, \dots, \pi_p, \pi_S$ de la chaîne de Markov cachée, il permet de rechercher la séquence $\boldsymbol{s}^\star \in J^n$ des états cachés dont la distribution conditionnelle à la séquence observée $\boldsymbol{x}$ est optimale. Formellement :

$$s^\star(\boldsymbol{x}) = \underset{\boldsymbol{s}\in J^n}{\operatorname{argmax}} \mathbb{P}(\boldsymbol{x}, \boldsymbol{s})$$

Cet algorithme procède par programmation dynamique. En effet, pour tout $t \in \{1, \dots, n\}$, notons $v_t(j)$ la probabilité du chemin caché le plus probable et terminant dans l'état j en position t :

$$\forall\, t \in \{1, \dots, n\}, \forall\, j \in J, v_t(j) = \max_{\boldsymbol{s}\in J^t, s_t = j} \mathbb{P}(\boldsymbol{x}, \boldsymbol{s})$$

Cette quantité vérifié la relation de récurrence suivante.

LEMME 1. *Soit $\boldsymbol{x} \in \mathscr{A}^n$ la séquence observée d'une chaîne de Markov cachée à p états cachées et d'ordre k, de matrices de transitions $\pi_1, \dots, \pi_p, \pi_S$. Alors, pour tout $t \in \{1, \dots, n\}$ et tout état caché $j \in J$, la quantité $v_t(j)$ vérifie la relation :*

$$v_{t+1}(j) = \pi_j(\boldsymbol{x}_{t-k,t-1}, x_t) \max_{k\in J} v_t(k)\pi_S(k, j)$$

Puisque :

$$\forall\, t \in \{1, n-1\}, \forall\, j \in J, v_{t+1}(j) = \max_{\boldsymbol{s}\in J^{t+1}, s_{t+1}=j} \prod_{t'=1}^{t+1} \pi_{s_{t'}}(\boldsymbol{x}_{t'-k,t'-1}, x_{t'})\pi_S(s_{t'-1}, s_{t'})$$

et que le domaine du maximum est l'ensemble des chemins cachés jusqu'à la position $t+1$ terminant dans l'état j, on peut factoriser ainsi l'expression :

$$v_{t+1}(j) = \pi_j(\boldsymbol{x}_{t-k+1,t}, x_{t+1}) \max_{\boldsymbol{s}\in J^t} \pi_S(s_t, j) \prod_{t'=1}^{t} \pi_{s_{t'}}(\boldsymbol{x}_{t'-k,t'-1}, x_{t'})\pi_S(s_{t'-1}, s_{t'})$$

Enfin, on peut *séparer* les contributions de la position $t+1$ et du début de la séquence cachée de la manière suivante :

$$v_{t+1}(j) = \pi_j(\boldsymbol{x}_{t-k+1,t}, x_{t+1}) \max_{k \in J} \pi_S(k,j) \max_{\boldsymbol{s} \in J^t, s_t = k} \prod_{t'=1}^{t} \pi_{s_{t'}}(\boldsymbol{x}_{t'-k,t'-1}, x_{t'}) \pi_S(s_{t'-1}, s_{t'})$$

En reconnaissant $v_t(k)$ dans le dernier terme de l'équation précédente, il vient :

$$\forall t \in \{1,\dots,n\}, \forall j \in J, v_{t+1}(j) = \pi_j(\boldsymbol{x}_{t-k_j+1,t}, x_{t+1}) \max_{l \in J} \pi_S(l,j) v_t(l)$$

soit la relation de récurrence annoncée. □

Ce résultat montre que, moyennant une initialisation, la quantité $v_t(j)$ peut être calculée récursivement le long de la séquence, et ce pour toute valeur $j \in J$. L'initialisation est obtenue en considérant que toutes les séquences commencent en une position fictive d'indice 0, dans une état d'indice 0, et pour lequel $v_0(0) = 1$.

THÉORÈME 15 (Algorithme de VITERBI). *L'un des chemins les plus probables $s^\star(\boldsymbol{x}) \in J^n$ sachant la séquence observée $\boldsymbol{x} \in \mathscr{A}^n$, ainsi que la probabilité de chacune des solutions du problème d'optimisation, peuvent être calculés récursivement en fonction des paramètres de le chaîne de Markov cachée.*

Nous venons en effet de voir que $v_t(j)$ pouvait se calculer récursivement pour tous les états $j \in J$. Or, si l'on note $j^\star$ la quantité :

$$j^\star = \operatorname*{argmax}_{j \in J} v_n(j)$$

on a :

$$\mathbb{P}(s^\star(\boldsymbol{x}), \boldsymbol{x}) = v_n\left(j^\star\right)$$

Mais $v_n\left(j^\star\right)$ est la probabilité du couple formé de la séquence observée $\boldsymbol{x}$ et le chemin caché $\boldsymbol{s}'$ constitué des arguments des maxima calculés à chaque pas de la récursion :

$$\forall t \in \{1,\dots,n\}, s'_t = \operatorname*{argmax}_{j \in J} \pi_S(j, s'_{t+1}) v_t(j)$$

Si bien que la séquence $s' \in \operatorname*{argmax}_{\boldsymbol{s} \in J^n} \mathbb{P}(\boldsymbol{x}, \boldsymbol{s})$, autrement dit $\boldsymbol{s}'$ est bien un chemin caché parmi les chemins cachés les plus probables sachant la séquence observée. □

On remarque que cet algorithme réalise la segmentation de la séquence avec une complexité qui dépend linéairement de la longueur de la séquence d'une part, et du nombre d'états cachés d'autre part. Le premier point est particulièrement crucial dans les applications à la détection de gènes, pour lesquelles on soumet à l'algorithme des génomes complets, c'est-à-dire constitués typiquement de quelques millions de nucléotides.

2.2.2. *Calcul de la distribution sur le chemin caché sachant la séquence observée.* Comme remarqué précédemment, l'algorithme de Viterbi ne fournit que le mode de la distribution sur le chemin caché sachant la séquence observée. On peut s'intéresser, plus généralement, au calcul de cette distribution. Plus précisément, l'algorithme présenté ci-après permet le calcul de la distribution de probabilité sur les états cachés sachant la séquence observée en une position particulière de la séquence, $\mathbb{P}(S_t = j | \boldsymbol{X})$, $t = 1,\dots,n$ et $j \in J$.

Pour ce faire, on commence par calculer la probabilité $\mathbb{P}(\boldsymbol{x})$ de la séquence observée $\boldsymbol{x}$, le chemin caché étant inconnu. Cette quantité répond naturellement à la définition suivante :

$$\forall \boldsymbol{x} \in \mathscr{A}^n, \mathbb{P}(\boldsymbol{x}) = \sum_{\boldsymbol{s} \in J^p} \mathbb{P}(\boldsymbol{x}, \boldsymbol{s})$$

La solution gloutonne consistant à effectuer explicitement la sommation sur tous les chemins cachés n'est pas raisonnable, car il faudrait considérer p^n chemins cachés différents. D'autant qu'il existe une solution de complexité linéaire, qui consiste à reprendre l'algorithme de Viterbi, et à y substituer une somme au supremum.

En effet, pour une séquence observée $\boldsymbol{x} \in \mathscr{A}^n$ donnée, et pour tout $t = 1, \ldots, n$ et $j \in J$, notons :

$$f_t(j) = \mathbb{P}(\boldsymbol{x}_{1,t}, S_t = j)$$

La quantité $f_t(j)$ peut alors être calculée récursivement, puisque :

$$\forall t \in \{1, \ldots, n-1\}, \forall j \in J, f_{t+1}(j) = \pi_j(\boldsymbol{x}_{t-k_j}) \sum_{i \in J} \pi_S(i,j) f_t(i) \tag{25}$$

L'exploitation de cette récurrence donne lieu à l'algorithme *forward*, identique en tout point à l'algorithme de Viterbi, à l'exception de la formule définissant la récurrence.

La propriété intéressante de cet algorithme est que, pour obtenir la probabilité de la séquence observée, il suffit d'effectuer la somme sur les états cachés des derniers termes de la séquence, $\left(f_n(j)\right)_{j \in J}$.

PROPOSITION 2. *La probabilité d'une séquence $\boldsymbol{x} \in \mathscr{A}^n$ sous un modèle de chaîne de Markov cachée à p états cachés de paramètres $\pi_1, \ldots, \pi_p, \pi_S$ s'écrit :*

$$\mathbb{P}(\boldsymbol{x}) = \sum_{j \in J} f_n(j)$$

où pour tout $j \in J$, $f_n(j)$ désigne le n^e terme d'une suite initialisée par $f_0(0) = 1$ et $f_0(j) = 0$ pour $j \in J$, et définie par la relation de récurrence (25).

Disposant de la probabilité de la séquence observée $\boldsymbol{x} \in \mathscr{A}^n$, calculée la distribution sur les états cachés sachant celle-ci en une position $t \in \{1, \ldots, n\}$ requiert d'évaluer $\mathbb{P}(\boldsymbol{x}, S_t = j)$, pour $j \in J$. Or :

$$\forall t \in \{1, \ldots, n\}, \forall j \in J, \mathbb{P}(\boldsymbol{x}, S_t = j) = \mathbb{P}(\boldsymbol{x}_{1,t}, S_t = j) \mathbb{P}(\boldsymbol{x}_{t+1,n} | \boldsymbol{x}_{1,t}, S_t = j)$$

soit encore, en prenant en compte la propriété de Markov :

$$\forall t \in \{1, \ldots, n\}, \forall j \in J, \mathbb{P}(\boldsymbol{x}, S_t = j) = \mathbb{P}(\boldsymbol{x}_{1,t}, S_t = j) \mathbb{P}(\boldsymbol{x}_{t+1,n} | \boldsymbol{x}_{t-\max(k_j),t}, S_t = j)$$

Le premier terme a dores et déjà été calculé par l'algorithme *forward*. Il reste donc à calculer le second, ce que l'on effectue par une récurrence arrière, cette fois, donnant lieu à l'algorithme *backward*.

En effet, si l'on note $b_t(j)$ la quantité $\mathbb{P}(\boldsymbol{x}_{t+1,n} | \boldsymbol{x}_{t-\max(k_j),t}, S_t = j)$, on vérifie alors que :

$$\forall t \in \{1, \ldots, n-1\}, \forall j \in J, b_t(j) = \sum_{i \in J} \pi_S(j,i) \pi_i(\boldsymbol{x}_{t-k_i+1,t}, x_{t+1}) b_{t+1}(i) \tag{26}$$

Muni des valeurs des variables forward (25) et backward (26) le long de la séquence, et pour chacun des états cachés, il est alors possible d'évaluer la probabilité d'un état caché en une position t sachant la séquence observée :

$$\forall t \in \{1, \ldots, n\}, \forall j \in J, \mathbb{P}(S_t = j | \boldsymbol{x}) = \frac{f_t(j) b_t(j)}{\mathbb{P}(\boldsymbol{x})}$$

2.2.3. *Estimation.* Jusqu'à présent, les différentes méthodes que nous avons présentées sur les chaînes de Markov cachées supposaient les paramètres de ce modèle (distribution d'émission et matrice de transition de la chaîne cachée) connus. Cependant, il n'est pas nécessaire de disposer d'une estimation supervisée de ces paramètres. On entend ici par *supervisé* le fait d'utiliser une ou plusieurs séquences dont la segmentation serait connue pour estimer le modèle (ce qui, dans ce cas, revient à estimer $p+1$ chaînes de Markov de manière classique).

En effet, l'algorithme de BAUM-WELCH permet l'estimation des paramètres d'une chaîne de Markov cachée en ne disposant que de la séquence observée. Il s'appuie pour cela sur les algorithmes présentés précédemment, les employant de manière itérative.

Le principe de l'algorithme consiste à calculer, sous les paramètres courants, l'espérance (par rapport au chemin caché) du nombre d'occurrences de chaque transition entre états cachés possible. Formellement, la méthode s'appuie sur la relation suivante :
(27)

$$\forall t \in \{1,\dots,n-1\}, \forall (i,j) \in J^2, \mathbb{P}(S_t = i, S_{t+1} = j|\boldsymbol{x}) = \frac{f_t(i)\pi_S(i,j)\pi_j(\boldsymbol{x}_{t-k_j+1,t}, x_{t+1})b_{t+1}(j)}{\mathbb{P}(\boldsymbol{x})}$$

où les distributions de probabilité considérées sont celles issues de l'itération précédente de l'algorithme. Le nombre attendu $A_{i,j}$ de transitions d'un état i vers un état j le long de la séquence est alors obtenu par :

$$A_{i,j} = \sum_{t=1}^{n-1} \mathbb{P}(S_t = i, S_{t+1} = j|\boldsymbol{x})$$

De la même manière, le nombre attendu $A_j(\boldsymbol{w}_{1,k_j+1})$ d'occurrences du mot $\boldsymbol{w}_{1,k_j+1}$ de longueur k_j+1 terminant dans l'état caché j vérifie :

$$A_j(\boldsymbol{w}_{1,k_j+1}) = \sum_{t|\boldsymbol{x}_{t-k_j,t}=\boldsymbol{w}_{1,k_j+1}} f_t(j)b_t(j)$$

Utilisées comme des comptages, ces quantités attendues permettent d'estimer les paramètres de la chaîne de Markov cachée qui seront utilisés pour l'itération suivante.

L'algorithme de BAUM-WELCH présente une propriété particulière : la probabilité de la séquence observée $\boldsymbol{x}$, sous les paramètres courants, augmente à chaque itération de l'algorithme. Cet argument est fondamental pour établir la convergence de l'algorithme vers une valeur hypothétique des paramètres de la chaîne de Markov, ou en d'autres termes que le résultat de cet algorithme appoche l'estimateur du maximum de la *vraisemblance incomplète*. Il est par ailleurs établi que cet estimateur est consistant.

Cet algorithme est essentiel pour les applications à la détection de gènes, car il signifie que la segmentation peut être effectuée au moyen de paramètres estimés sur le génome à segmenter directement. Et ce sans recourir, comme dans le cas de l'approche par classification, à l'approximation consistant à entraîner le modèle codant sur les plus grands cadres ouverts de lecture identifiés dans le génome. Cette approximation peut en effet produire des prédictions erronées de la position du codon `START` d'un gène.

2.3. Les modèles utilisés en détection de gènes. Les modèles utilisés pour la détection de gènes permettent de prendre en compte, d'une manière globale, l'ensemble des signaux connus comme étant associés à la présence d'un gène : le biais d'usage des codons, mais aussi le site de fixation du ribosome, voire le site d'initiation de la transcription. La figure 3 décrit graphiquement le graphe de transitions entre les états cachés utilisés de manière standard par le logiciel SHOW, développé par P. Nicolas [**?**], et destiné à la détection des gènes dans un génome procaryote. Le modèle de Markov caché associé à ce graphe de transitions permet la reconnaissance du site de fixation du ribosome, ainsi que la détection de gènes chevauchants, voire partageant quelques nucléotides entre leurs codons `START` et `STOP` respectifs.

Ce graphe de transitions peut être complexifié à volonté, de manière à intégrer l'ensemble des signaux que l'utilisateur souhaite exploiter pour préciser la position des gènes. Notons également que ces modèles présentent une flexibilité suffisante, et bénéficient d'algorithmes suffisamment performants, pour être également utilisés pour la prédiction des gènes dans les génome d'organismes eucaryotes. Ces derniers présentent un certain nombre de caractéristiques qui rendent pourtant cette tâche beaucoup plus difficile, tels que les décalages de phase (les séquences codantes n'ont alors plus des longueurs multiples de trois) ou les phénomènes d'épissage (des portions de gènes, flanqués ni de codon `START` ni de codon `STOP`, ne sont pas codantes, et par conséquent ne présentent pas de biais d'usage du code génétique).

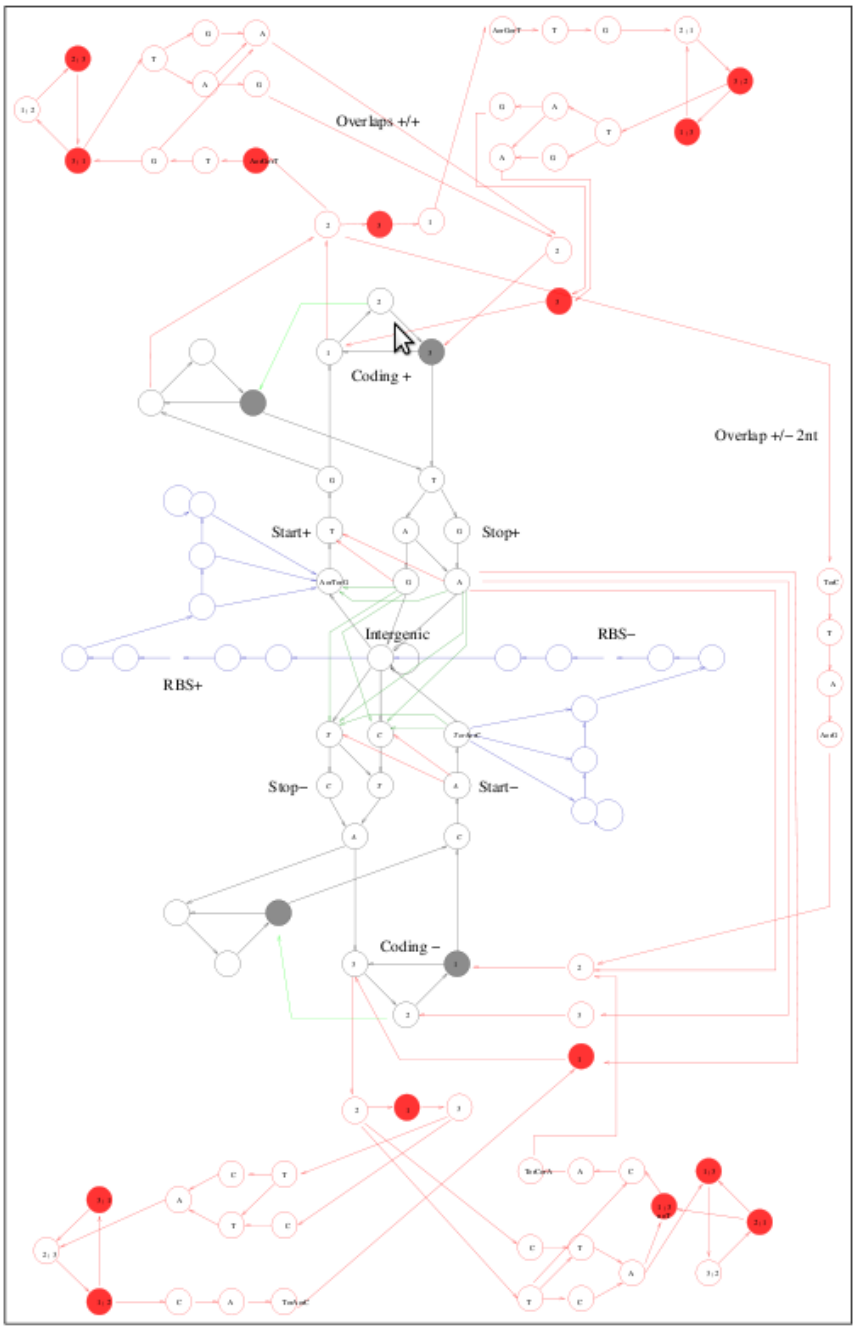


FIG. 3. Exemple de spécification des transitions entre états cachés pour une chaîne de Markov cachée destinée à la détection de gènes. Cette spécification permet la reconnaissance du signal de fixation du ribosome en amont des séquences codantes, ainsi que la prédiction de gènes chevauchants.

3. Prédiction de fonction

En aval de la détection de gènes, les séquences codantes identifiées dans un génome sont *annotées*, au sens où l'on cherche à leur attribuer une fonction. Selon les types d'activité que peuvent avoir les protéines, diverses nomenclatures : dans le cas des enzymes, une nomenclature permet d'exprimer de manière détaillée les réactions chimiques qu'elles peuvent catalyser, spécifiant également les éventuels cofacteurs mis en jeu par cette catalyse ; dans le cas des protéines essentielles à l'expression de l'information génétique (sous-unités de l'ARN-polymerase, du ribosome,...), le seul nom de la protéine suffit à décrire sa fonction. Mais pour de nombreuses autres, c'est essentiellement un paragraphe de texte qui décrit la fonction.

Indépendamment des questions de nomenclature des fonctions, il faut garder à l'esprit que la prédiction de fonction d'une protéines s'appuie systématiquement sur la comparaison de sa séquence à des séquences accumulées dans des bases de données, accompagnées éventuellement des informations relatives à leur fonction. Cette comparaison peut être conduite de diverses manières, en utilisant des outils plus ou moins structurés et élaborés : les plus élémentaires (mais aussi les plus utilisés) s'appuient sur une comparaison deux-à-deux de la séquence à annoter avec les séquences de la base de données, et les plus élaborées évaluent la vraisemblance de la séquence à annoter contre un modèle de chaîne de Markov cachée décrivant la succession des domaines de la protéine le long de sa séquence. Dans ce dernier cas, les modèles de chaîne de Markov cachée employés sont entraînés sur des ensembles de séquences très similaires, et partageant la même fonction.

Quelque soit la méthode employée pour quantifier la similarité entre la séquence à annoter et la ou les séquences connues pour coder les protéines remplissant un rôle donné dans le fonctionnement de la cellule, il est crucial de savoir quantifier la significativité de la similarité entre la séquence à annoter et la ou les séquences "exemples" d'un rôle fonctionnel. Ce problème est de nature statistique, et met également en jeu les chaînes de Markov.

La prédiction de structure est une approche assez radicalement différente de la prédiction de fonction ambitionne de dépasser la seule généralisation de la connaissance permise par les méthodes de comparaison de séquences. Cette ambition est justifiée par l'émergence de technologies de séquençage telles que la métagénomique, qui visent à identifier les séquences de nouvelles enzymes d'intérêt industriel et environnemental dans des organismes potentiellement incultivables. La masse de données générées par ces technologies, de même que la vitesse croissante de séquençage de nouveaux organismes, alimente un stock toujours croissant d'enzymes dont les fonctions sont inconnues, et qui ne présentent pas de similarité nette avec aucune enzyme connue. Les protéines, bien que constituées d'une succession linéaire d'acides aminés, se replient de manières complexes de manière à positionner leurs sites actifs au contact des substrats avec lesquels ils doivent interagir. Aussi, la connaissance de la structure d'une protéine est une information essentielle à la compréhension de son interaction avec ses partenaires réactionnels.

3.1. Significativité des alignements. L'évaluation de la similarité entre deux séquences protéiques se heurte à un obstacle non négligeable : sachant que deux protéines héritées d'un même ancêtre chez deux organismes différents peuvent différer en taille suite à la délétion ou l'insertion de quelques acides aminés, comparer les séquences de deux protéines en évaluant un score de similarité position par position, en commençant par les premières lettres des deux séquences, ne peut suffire. Il est en fait

nécessaire d'inférer les positions qui ont pu être perdues dans l'une ou l'autre des séquences, pour pouvoir ensuite calculer un score de similarité. En fait, cette tâche d'inférence des positions perdues par l'une ou l'autre des séquences est couplée à l'évaluation de la similarité des séquences, et donne lieu à un exercice appelé *alignement de séquences*.

Divers algorithmes permettant de réaliser l'alignement de deux séquences ont été proposés depuis l'émergence de ce besoin dans les années 80. Mais tous s'appuient sur le même principe : inférer les positions perdues de manière à optimiser la similarité des séquences. Le plus simple des algorithmes permettant de réaliser ce travail est l'algorithme de NEEDLEMAN et WUNSCH. Il permet de trouver l'alignement optimal couvrant toute la longueur de chacune des séquences. L'optimalité s'entend ici comme la maximisation d'un score, qui résulte de la sommation le long de l'alignement d'un score reflétant la fréquence (préalablement calculée avec un ensemble d'entraînement) de la mutation entre deux acides aminés lorsqu'il ne s'agit pas de positions perdues dans l'une des séquences, et d'un coût lorsqu'une position, ou un segment, dans le cas contraire.

Une variante de cet algorithme, l'algorithme de SMITH et WATERMAN, permet de rechercher l'alignement de score le plus élevé en ôtant la contrainte de couvrir l'ensemble des deux séquences. Implémentable avec une complexité linéaire en le produit des longueurs des séquences à aligner, ces algorithmes ne sont pas utilisables pour comparer une séquence à une base de données qui en contient plusieurs centaines de milliers, parfois plus. On utilise dans ce cas des algorithmes heuristiques, qui permette de retrouver avec une assez bonne certitude les séquences les plus ressemblantes à la séquence à annoter, sans pour autant effectuer l'évaluation exacte du score de l'alignement optimal.

Quelque soit la méthode employée, rien n'assure que la séquence la plus ressemblante à la séquence à annoter assure la même fonction. Pour en juger, il faut pouvoir évaluer la significativité du score de l'alignement optimal entre les deux séquences, par exemple en calculant la significativité du rapport de vraisemblance entre le modèle de mutation (qui traduit l'existence d'un ancêtre commun aux deux protéines) et un modèle d'indépendances des séquences. Si les modèles d'indépendance des positions sont très fréquemment employés pour ce faire, il est également usuel de recourir à des modèles de Markov. La composition locale en acides aminés des protéines est en effet biaisée d'une manière indépendante de leur fonction, et la prise en compte de ce phénomène par les chaînes de Markov contribue à affiner l'évaluation de la significativité des alignements.

3.2. Prédiction de structure des protéines. La structure d'une protéine est formellement définie par les coordonnées spatiales relatives de chacun des atomes qui la composent. Cependant, ces structures sont abstraites sous diverses formes, plus ou moins détaillées, mais qui visent systématiquement à recoder à différentes échelles des motifs structurels récurrents.

En premier lieu, on distingue des motifs récurrents dans les structures protéiques : localement, les protéines s'organisent en régions quasiment planes, les *feuillets*, ou bien en *hélices*. Ces deux types de structure sont reliés par des segments sans structure identifiable. Attribuer chacun des acides aminés d'une protéine à l'une de ces structures définit la *structure secondaire* de la protéine. Un modèle plus élaboré consiste à identifier quelques dizaines de motifs élémentaires, qui forment les *alphabets structuraux*(voir [**?**]), à partir desquels on sait prédire la structure de la protéine. Le leitmotiv de ces approches est qu'en permettant une reformulation de la séquence de la protéine en utilisant un nombre de motifs très inférieurs à la combinatoire des séquences d'acides aminés correspondantes, elles rendent possibles la prédiction de la structure, qui, elle, met

en jeu des interactions de longue portée, et demande donc d'explorer un grand nombre de combinaisons des motifs.

Prédire la structure secondaire d'une protéine *de novo*, c'est-à-dire à partir de la seule connaissance de la séquence de la protéine, est par conséquent un enjeu important en direction de la prédiction de la structure des protéines à partir de la séquence. L'utilisation des chaînes de Markov cachées à cette fin, chaque type de structure secondaire étant représenté par un ou plusieurs des états cachés, permet en effet de segmenter de nouvelles protéines en utilisant l'algorithme de VITERBI.

Sans approcher les qualités de prédiction des algorithmes exploitant les structures connues de protéines proches, cette approche *ab initio*, développée par Juliette MARTIN, présente des performances significatives [**?**].

Deuxième partie

Compression de texte

Introduction

Dans les années 1830, le premier télégraphe électrique fut inventé par Samuel MORSE et Alfred VAIL. Il s'agissait de pouvoir transmettre des *messages*, au sens de séquences de caractères, en utilisant l'alternance des deux signaux que le télégraphe permettait de transmettre : le signal court, symbolisé par `.`, et le signal long, symbolisé par le `−`. Destiné à être interprété par un opérateur humain, ce code binaire n'était pas utilisé directement pour coder le message. Chacun de ces signaux était plutôt lui-même représenté de manière unique comme une séquence d'impulsions cadencées par un pas de temps arbitraire : un `.` correspond à une excitation d'une durée d'une unité de temps, alors qu'un `−` correspond à une excitation d'une durée de trois unités de temps.

Depuis lors, l'arrivée des équipements électroniques et des ordinateurs, suivie des supports de stockage de l'information et des réseaux ont placé les questions liées à la représentation binaire de l'information au cœur de nombreuses technologies. Ces questions ont motivé le développement d'une théorie mathématique, la *théorie de l'information*, qui s'attachent à décrire les propriétés des méthodes de représentation binaire de l'information. Cette partie résume les grandes lignes de cette théorie, reprenant dans un premier temps ses fondements pour ensuite aborder les questions statistiques qui en dérivent lorsque l'on cherche à minimiser la longueur attendue d'un message codé.

Le chapitre qui suit introduit le formalisme du codage, puis détaille les résultats fondamentaux de la théorie de la compression. Ensuite, le chapitre 2 détaillera le parallèle entre la théorie de la compression et le principe de maximum d'entropie, pour ensuite discuter du choix de la loi de codage en fonction d'un échantillon à transmettre. Le chapitre 3 sera lui consacré aux codes adaptatifs, pour lesquels une loi de codage est recalculée après la réception de chaque symbole, et sera l'occasion de retrouver les fondements de la statistique bayésienne à partir du principe de maximum d'entropie. Enfin, les chapitres 4 et 5 seront consacrés à l'exploitation des dépendances statistiques locales du message pour affiner la loi de codage.

CHAPITRE 4

Codes, loi de codages et compression

1. Codes

Pour s'échanger un message textuel écrit avec l'alphabet usuel à 26 lettres en utilisant un télégraphe, l'émetteur du message et son récepteur partagent une table de correspondance entre chacune des 26 lettres de l'alphabet et une séquence de 0 et de 1 que l'on appelle un *code*.

DÉFINITION 8. *Soit $\mathscr{A}$ un alphabet fini. Une application $c : \mathscr{A} \to \{0,1\}^\star$, injective, est appelée un* code binaire *pour l'alphabet $\mathscr{A}$.*

Le caractère injectif de cette application est essentiel. Le receveur du message doit en effet pouvoir inverser l'application c pour décoder le message.

Mais en général, un message est composé de plusieurs symboles successifs, et non d'un seul. On code alors les symboles successifs du message selon le code c défini pour chacun d'eux. On parle alors de *codage séquentiel.*

DÉFINITION 9. *Soit c un code pour l'alphabet $\mathscr{A}$. Le code $C : \mathscr{A}^\star \to \{0,1\}^\star$ défini par :*

$$\forall \boldsymbol{x} \in \mathscr{A}^\star, C(\boldsymbol{x}) = c(x_1)\dots c(x_{|\mathscr{A}|})$$

où $c(x_1)\dots c(x_{|\mathscr{A}|})$ désigne le concaténation des séquences binaires $c(x_i), i = 1,\dots,|\mathscr{A}|$, est appelé code séquentiel *associé à c.*

Pour vérifier que le code séquentiel C associé au code c est bien un code pour l'alphabet $\mathscr{A}^\star$, il est nécessaire de vérifier que C est bien injective. Or, il est clair que qu'en toute généralité, le code séquentiel dérivé d'un code n'est pas injectif. Selon le code Morse (fig. 1), par exemple, le message `MORSE` s'écrit `−−−.−.....`. Comme `M` et `O` sont codés avec un nombre variable de symboles identiques, respectivement − et `−−`, les cinq premiers bits de la séquence codée peuvent être interprétés de deux manières : comme `MO`, mais aussi comme `OM`. Quand bien même la suite de la séquence pourrait être décodée de manière non ambiguë comme `RSE`, il resterait tout de même deux séquences, `MORSE` et `OMRSE`, dont les images par le code séquentiel sont identiques.

Pour contourner ce problème et assurer l'injectivité du code séquentiel, le code MORSE prévoit un troisième symbole, que l'on note ici $\star$, transmis sous la forme de trois unités de temps sans signal, pour séparer les séquences de bit correspondant à chacun des symboles. De la même manière, les espaces séparant les mots du message original sont codés par une séquence de sept unités de temps sans signal. Ainsi, le code MORSE séquentiel C encode le message $\boldsymbol{x} \in \mathscr{A}^\star$ sous la forme :

$$C(\boldsymbol{x}) = c(x_1) \star \cdots \star c(x_{|\boldsymbol{x}|})$$

Comme d'après la convention de codage, aucune succession des bits − et `.` n'implique de pause de trois unités de temps ou plus, les occurrences de $\star$ dans la séquence $C(\boldsymbol{x})$ peuvent être immédiatement détectées. L'application de composition séquentielle du message, qui au vecteur $(c(x_1),\dots,c(x_{|\boldsymbol{x}|})$ associe la séquence $C(\boldsymbol{x})$, est dès lors injective, ce qui assure l'injectivité de l'application C elle-même.

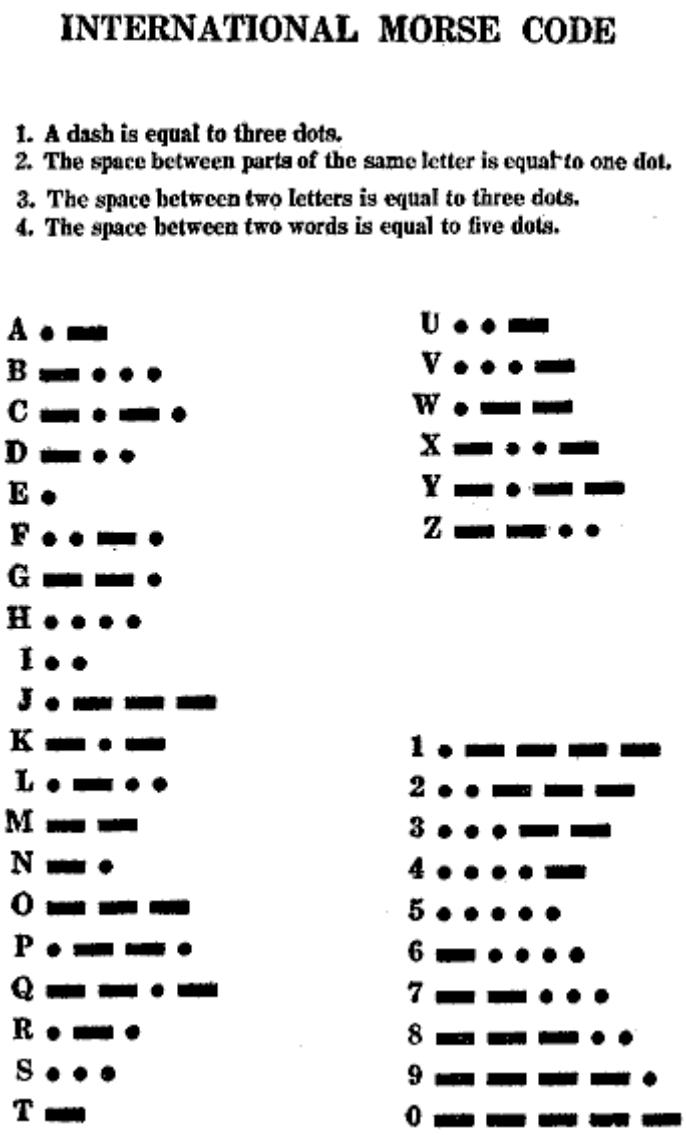


FIG. 1. Table de codage du code MORSE.

1.1. Automate de décodage. Comme remarqué précédemment, le caractère *spécifique* de la séquence de séparation $\star$ est essentiel au caractère injectif du code séquentiel. Par spécifique, on entend *qui puisse être décodé sans ambiguité*, ce qui requiert que $\star$ ne soit une sous-chaîne alignée à gauche d'aucun code $c(a), a \in \mathscr{A}$. On dit que la séquence $\star$ n'est le préfixe d'aucune séquence $c(a), a \in \mathscr{A}$.

DÉFINITION 10. *Soit* $\boldsymbol{b} \in \{0,1\}^\star$. *L'ensemble des sous-chaînes de* $\boldsymbol{b}$ *alignées à gauche est noté* pref($\boldsymbol{b}$), *et défini formellement par la relation :*

$$\text{pref}(b_1 \dots b_n) = \{b_1 \dots b_j, j \leq n\} \cup \{\emptyset\}$$

Un élément $\boldsymbol{b}'$ *de* pref($\boldsymbol{b}$) *est appelé un* préfixe *de* $\boldsymbol{b}$.

Mais en approfondissant cette notion, on s'aperçoit qu'elle permet également de définir des codes qui ne requièrent pas le recours à un symbole de terminaison des

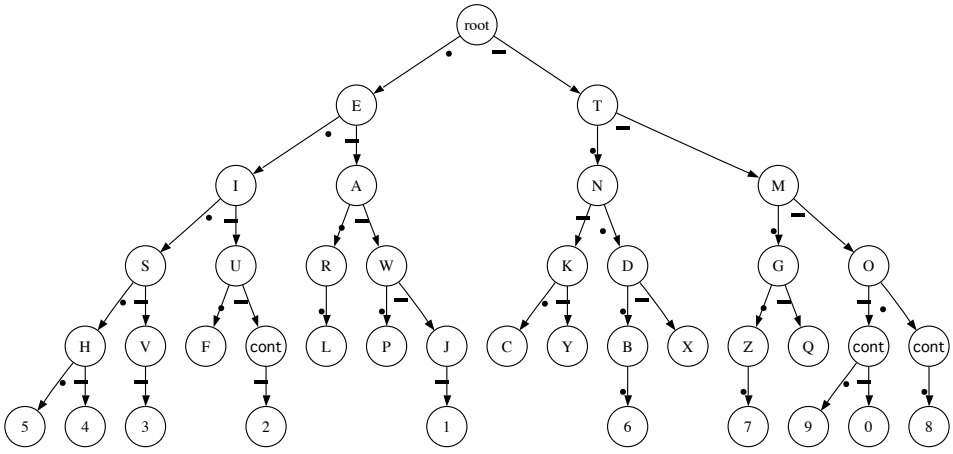
séquences de bits correspondant à chacun des symboles du message. Pour illustrer le propos, nous envisageons le cas d'un automate $\mathbb{A}(c)$ qui décoderait tout message codé selon c, que nous décrivons à présent.

A partir d'un état initial noté $\emptyset$, l'automate peut transiter dans l'état 0 ou 1 selon le bit reçu, puis dans un état parmi $\{0,1\}^2$, et ainsi de suite jusqu'à avoir atteint un état permettant la reconnaissance d'un symbole du message. Formellement, on considère que l'espace d'état de l'automate est $\{0,1\}^{\max_a |c(a)|}$, et l'on distingue les états transitoires et finaux, qui forment les états admissibles pour l'automate, des états invalides. Pour ce faire, chacun de ces états porte un label $l(\boldsymbol{b})$ défini par :

$$\forall \boldsymbol{b} \in \cup_{k=0}^{\max_{a\in\mathscr{A}}|c(a)|}\{0,1\}^k, l(\boldsymbol{b}) = \begin{cases} \texttt{continue} & \text{si } \exists a \in \mathscr{A}, \boldsymbol{b} \in \text{pref}(c(a)) \\ a & \text{si } c(a) = \boldsymbol{b}, a \in \mathscr{A} \\ \texttt{invalide} & \text{sinon} \end{cases}$$

L'automate de décodage $\mathbb{A}(c)$ suit le système de transition suivant : partant de l'état initial $\emptyset$, il continue de lire les bits et changer d'état en conséquence tant que l'état atteint porte le label `continue`. Lorsque le label atteint est un caractère $a \in \mathscr{A}$, l'automate a reconnu un caractère et l'ajoute au message décodé, puis revient dans l'état $\emptyset$. S'il atteint le label `invalide`, le décodage du message échoue.

EXEMPLE 10.1. *L'automate de décodage pour le code Morse est représenté par le graphe d'états suivant :*



Ici, le système de transition est un peu particulier, compte-tenu du fait que les symboles . et − sont eux-mêmes encodés comme un signal d'une et de trois unités de temps de long respectivement. Lorsque l'automate parvient sur un noeud permettant la reconnaissance d'un symbole, il attend le signal suivant : s'il s'agit d'une pause d'au moins trois unités de temps, la reconnaissance du symbole a lieu, sinon la transition vers l'état enfant est effectuée.

1.2. Terminaison de l'automate. Cet automate ne fonctionnera correctement que si les états atteignables depuis un état permettant la reconnaissance d'un caractère portent le label `invalide`. Sinon, il existe un caractère a dont la reconnaissance implique de passer par un état permettant la reconnaissance d'un autre caractère b, et la reconnaissance de a n'aura jamais lieu. Comme un état $\boldsymbol{b}$ n'est atteignable que depuis l'ensemble de ses préfixes dans ce système de transition, cela signifie qu'il faut recourir à un code *préfixe*.

DÉFINITION 11. *Un code $c : \mathscr{A} \to \{0,1\}^\star$ est dit préfixe si le code d'aucun caractère n'est un préfixe du code d'un autre caractère, soit :*

$$\forall a \neq b \in \mathscr{A}^2, c(a) \notin \text{pref}(c(b))$$

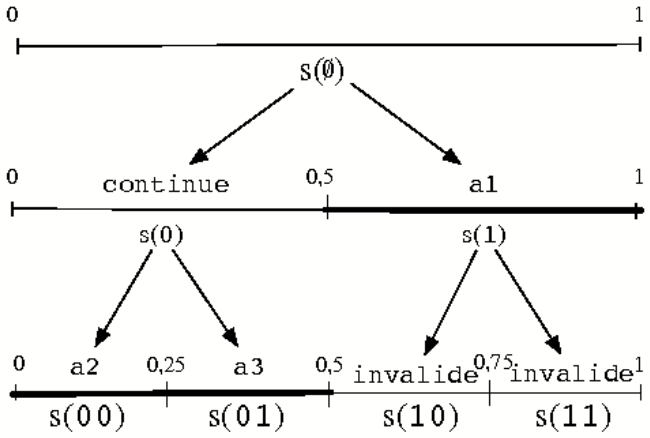


FIG. 2. Illustration du lien entre la dichotomie récursive de l'intervalle $[0,1[$ et un code binaire. Les traits en gras indiquent un code préfixe possible où un alphabet de symboles de taille 3 est codé sur deux bits au maximum, un symbole étant codé sur un bit.

On note $\mathscr{C}(\mathscr{A})$ l'ensemble des codes préfixes pour l'alphabet $\mathscr{A}$.

Le caractère *décodable* d'un tel code se formule ainsi :

THÉORÈME 16. *Un message $\boldsymbol{x}$ codé selon un code préfixe c peut être décodé par l'automate $\mathbb{A}(c)$ en un temps linéaire par rapport à la longueur du message $|\boldsymbol{x}|$.*

Il suffit de remarquer que, si la séquence $c(x_1,\dots,x_n)$ envoyée à l'automate est bien codée selon le code c, alors l'automate atteindra l'état $c(x_i)$, $i=1,\dots,n$. Comme $l(c(x_i))=x_i$, le symbole x_i sera reconnu. Pour décoder ainsi l'ensemble de la séquence de bits $c(\boldsymbol{x})$, le nombre de transition d'états $N(\mathbb{A}(c),\boldsymbol{x})$ effectué par l'automate est donc égal à $|c(\boldsymbol{x})|+|\boldsymbol{x}|$ (le second terme étant associé au retour à l'état $\emptyset$ après la reconnaissance de chacun des symboles du message). Comme l'alphabet $\mathscr{A}$ des symboles composant le message est fini, $|c(\boldsymbol{x})|$ et $|x|$ sont équivalents, et il existe un entier k tel que :

$$\max_{\boldsymbol{x}\in\mathscr{A}^n} N(\mathbb{A}(c),\boldsymbol{x}) \leq kn$$

□

REMARQUE 1. *Le code* MORSE *n'est donc pas un code préfixe : de nombreux nœuds de l'arbre sont associés à un symbole, et de même pour leurs descendants. Par exemple, le code* `Morse` *pour le symbole* `E` *est le préfixe des codes pour 17 autres symboles.*

En revanche, si l'on reformule le système de transition en termes de 4 signaux possibles (`.`,`-`,`finSymbole` - qui est codé par une pause de 3 unités de temps, et `finMot` *- qui est codé par une pause de 5 unités de temps), alors chaque noeud permettant la reconnaissance d'un symbole sera reconnu par le noeud enfant associé à la transition* `finSymbole`.

1.3. Lien avec la dichotomie récursive de $[0,1[$. L'automate de décodage introduit précédemment est un arbre binaire d'états que l'on peut utilement représenter par un arbre de dichotomie de l'intervalle $[0,1[$. Les dichotomies successives de l'intervalle $[0,1[$ se représentent également comme un arbre binaire, dont la racine est l'intervalle $[0,1[$ lui-même, et chaque noeud possède les deux intervalles issus de sa dichotomie pour descendants. Chacun des noeuds de cet arbre est un intervalle inclus dans $[0,1[$ qui peut être identifié par une séquence de bits de longueur égale à sa profondeur dans l'arbre. En effet, on peut interpréter une séquence de bits comme une séquence

d'instructions `moitié inférieure`/`moitié supérieure` régissant le parcours de l'arbre : `0` correspond à $[0;0,5[$, `00` à $[0;0,25[$, `001` à $[0,125;0,25[$, et ainsi de suite (voir figure 2).

Plus généralement, une séquence de bits $\boldsymbol{b}$ est associée à un intervalle $s(\boldsymbol{b}) = [s_{\min}(\boldsymbol{b}), s_{\max}(\boldsymbol{b})[\subset [0,1[$ tel que :

$$s_{\min}(\boldsymbol{b}) = \sum_{i=1}^{|\boldsymbol{b}|} \frac{b_i}{2^i} \qquad s_{\max}(\boldsymbol{b}) = s_{\min}(\boldsymbol{b}) + \frac{1}{2^{|b|}}$$

L'application s est injective en vertu de l'unicité de la décomposition binaire des bornes des intervalles. Le principal intérêt de cette représentation réside dans le théorème suivant :

THÉORÈME 17. *Soient deux séquences de bits $\boldsymbol{b}$ et $\boldsymbol{b}'$. Les deux propositions suivantes sont équivalentes :*

$$\boldsymbol{b}' \in \text{pref}(\boldsymbol{b}) \Leftrightarrow s(\boldsymbol{b}) \subset s(\boldsymbol{b}')$$

De plus, $s(\boldsymbol{b}) \cap s(\boldsymbol{b}') \neq \emptyset$ si et seulement si $\boldsymbol{b} \in \text{pref}(\boldsymbol{b}')$ ou $\boldsymbol{b}' \in \text{pref}(\boldsymbol{b})$.

En effet, si $\boldsymbol{b}' \in \text{pref}(\boldsymbol{b})$, on a de manière équivalente que :

$$s_{\min}(\boldsymbol{b}) = s_{\min}(\boldsymbol{b}') + \sum_{i=|\boldsymbol{b}'|+1}^{|\boldsymbol{b}|} \frac{b_i}{2^i} \geq s_{\min}(\boldsymbol{b}')$$

et de même que

$$s_{\max}(\boldsymbol{b}) = s_{\max}(\boldsymbol{b}') + \sum_{i=|\boldsymbol{b}'|+1}^{|\boldsymbol{b}|} \frac{b_i}{2^i} - \frac{1}{2^{|\boldsymbol{b}'|}} + \frac{1}{2^{|\boldsymbol{b}|}} \leq s_{\max}(\boldsymbol{b}')$$

ce qui démontre le premier point.

Pour démontrer le second point, on suppose que $\boldsymbol{b}' \in \text{pref}(\boldsymbol{b})$, et on considère $\text{pref}(\boldsymbol{b}, \boldsymbol{b}')$ le plus long préfixe commun à $\boldsymbol{b}$ et $\boldsymbol{b}'$. $\text{pref}(\boldsymbol{b}, \boldsymbol{b}')$ correspond également au plus récent ancêtre commun des noeuds associés à $\boldsymbol{b}$ et $\boldsymbol{b}'$. Comme l'arbre de dichotomie est binaire, soit les noeuds associés à $\boldsymbol{b}$ et $\boldsymbol{b}'$ descendent de chacun des enfants de $\text{pref}(\boldsymbol{b}, \boldsymbol{b}')$, soit l'ancêtre commun est l'un des noeuds $\boldsymbol{b}$ ou $\boldsymbol{b}'$. Le premier cas est exclus puisque l'intersection des intervalles associés à $\boldsymbol{b}$ et $\boldsymbol{b}'$ est non-vide, alors que les deux intervalles enfants de $\text{pref}(\boldsymbol{b}, \boldsymbol{b}')$ sont disjoints. Reste donc le second cas, qui équivaut à l'assertion $\boldsymbol{b} \in \text{pref}(\boldsymbol{b}')$ ou $\boldsymbol{b}' \in \text{pref}(\boldsymbol{b})$. La réciproque est immédiate. □

Ce théorème se traduit immédiatement au cas d'un code.

COROLLAIRE 17.1. *Les intervalles $\{s(c(a)), a \in \mathscr{A}\}$ associés aux images $\{c(a), a \in \mathscr{A}\}$ d'un code préfixe c ont une intersection vide deux-à-deux.*

La preuve de ce théorème est une conséquence immédiate du théorème précédent. Il signifie que choisir un code préfixe, c'est choisir un ensemble de noeuds dans l'arbre de dichotomie de $[0,1[$ de telle manière qu'aucun d'entre eux n'est un descendant d'un autre noeud, et associer à chacun de ces noeuds un caractère de l'alphabet à coder.

1.4. Construction de codes préfixes. Il est intuitif qu'un ensemble de noeuds préfixe-libre ne peut pas contenir un grand nombre de noeuds de l'arbre situés trop près de sa racine : il n'y en a simplement pas assez. Cette contrainte se traduit formellement par l'inégalité de KRAFT, dont la formulation suit.

THÉORÈME 18 (Inégalité de KRAFT). *Soit c un code préfixe sur l'alphabet fini $\mathscr{A}$. La relation suivante est toujours vérifiée :*

$$\sum_{a \in \mathscr{A}} 2^{-|c(a)|} \leq 1$$

Remarquons tout d'abord que pour $a \in \mathscr{A}$, $2^{-|c(a)|}$ désigne la longueur de l'intervalle associé à $c(a)$ dans la dichotomie de l'intervalle unité. Or, les intervalles associés à chacune des images de c dans la dichotomie de l'intervalle unité sont deux-à-deux disjoints et inclus dans l'intervalle unité : la somme de leurs longueurs est donc inférieure à 1. □

Mais la force de l'inégalité de KRAFT réside dans le fait que le théorème précédent admet une réciproque : quelles que soient les longueurs de code ciblées, il est possible de construire un code préfixe permettant de s'y conformer.

THÉORÈME 19 (Existence de codes préfixes). *Soit une longueur de code $l : \mathscr{A} \to \mathbb{N}$ vérifiant l'inégalité de* KRAFT *: $\sum_{a\in\mathscr{A}} 2^{-l(a)} \leq 1$. Alors il existe un code préfixe c pour l'alphabet $\mathscr{A}$ dont les longueurs de code $(|c(a)|)_{a\in\mathscr{A}}$ coïncident avec celles spécifiées : $|c(a)| = l(a)$.*

Soit donc une longueur de code l vérifiant l'inégalité de KRAFT. Ordonnons les symboles de $\mathscr{A}$ par ordre décroissant de longueur de code : $\mathscr{A} = \{a_i, l(a_i) > l(a_{i+1}), i = 1,\dots,|\mathscr{A}|-1\}$. A chaque symbole a_i de l'alphabet $\mathscr{A}$, associons l'intervalle $S(a_i)$ de $[0,1[$ tel que :

$$S(a_i) = \begin{cases} [0, 2^{-l(a_1)}[& \text{si } i = 1 \\ [\sum_{k=1}^{i-1} 2^{-l(a_k)}, \sum_{k=1}^{i} 2^{-l(a_k)}[& \text{si } 1 < i \leq |\mathscr{A}| \end{cases}$$

Pour chacun de ces intervalles $S(a_i)$, de longueur $2^{-l(a_i)}$, les bornes admettent un développement diadique de longueur $l(a_i)$ (c'est à dire s'écrivent sous la forme $\sum_{k=1}^{l(a_i)} b_k 2^{-k}$). On peut donc associer chacun d'eux à un noeud $\boldsymbol{b}(a_i)$ de l'arbre de dichotomie. Comme tous ces intervalles sont disjoints deux-à-deux, l'ensemble des séquences de bits $\{\boldsymbol{b}(a_i), i = 1,\dots,n\}$ forme un ensemble préfixe, et définit le code préfixe $c : a \to \boldsymbol{b}(a)$. Par ailleurs, le code associé à chaque symbole a de l'alphabet vérifie $|\boldsymbol{b}(a)| = -\log_2 2^{-l(a)} = l(a)$. □

1.5. Comparaison de codes. La fonction $\sum_{a\in\mathscr{A}} 2^{-|c(a)|}$ est strictement décroissante en chacun des arguments $|c(a)|, a \in \mathscr{A}$. L'inégalité de KRAFT pose donc fondamentalement une limite inférieure à la fonction de longueur de code $|c|$ d'un code préfixe c, et donc à sa capacité à transmettre un message en recourant à une quantité limitée de bits.

1.5.1. *Relation d'ordre partiel.* Si l'on souhaite minimiser le coût de transmission d'un message codé, il est évidemment préférable d'éviter tout *gâchis* de bits dans le choix du code, ce qui, d'après la remarque précédente, revient à saturer l'inégalité de KRAFT. En d'autres termes, on préférera toujours recourir à un code qui utilise un bit de moins pour encoder un symbole, tous les autres symboles restant encodés par un nombre de bits identiques. Plus généralement, on peut définir une relation d'ordre partielle sur les codes de la manière suivante.

DÉFINITION 12. *Soit deux codes c et c' sur l'alphabet $\mathscr{A}$. On dit que c est* plus court *que c', et l'on note $c \preceq c'$, si :*

$$\forall a \in \mathscr{A}, |c(a)| \leq |c'(a)|$$

S'il existe $a \in \mathscr{A}$ tel que $|c(a)| < |c'(a)|$, alors c est dit strictement plus court *que c', et on note $c \prec c'$.*

La relation $c \prec c'$ est une relation d'ordre partielle sur l'ensemble des codes (et en particulier sur l'ensemble des codes préfixes), qui permet de formaliser la préférence pour un code court.

1.5.2. *Codes complets.* En particulier, on peut s'intéresser à l'ensemble des codes préfixes les plus courts au sens de la relation d'ordre introduite ci-dessus. Comme nous l'avons vu précédemment, l'inégalité de KRAFT impose une borne inférieure à la longueur des codes préfixes. Nous allons voir à présent que l'inégalité de KRAFT peut toujours être saturée en un certain sens. Définissons tout d'abord la notion de *code complet*, qui traduit le fait qu'un code sature cette inégalité.

DÉFINITION 13. *Un code préfixe c sur un alphabet fini $\mathscr{A}$ est dit complet s'il vérifie la relation :*

$$\sum_{a\in\mathscr{A}} 2^{-|c(a)|} = 1$$

Il est évident de remarquer qu'un code complet est minimal pour l'ordre partiel $\prec$. Mais l'intérêt de cette notion provient plutôt de la réciproque de cette propriété, à savoir que tout code préfixe minimal pour la relation $\prec$ est complet, ce qu'exprime le théorème suivant.

THÉORÈME 20. *Soit c un code préfixe, alors il existe un code complet c_{min} tel que $c_{min} \prec c$.*

Si c est déjà un code complet, alors le résultat est acquis. Considérons donc un code c tel que l'inégalité de KRAFT ne soit pas saturée, et notons ε la quantité :

$$\varepsilon = 1 - \sum_{a\in\mathscr{A}} 2^{-|c(a)|}$$

Soit a_1 l'élément (ou l'un des éléments) de l'alphabet dont le code est le plus court et tel que $2^{-|c(a)|} < \varepsilon$:

$$a_1 \in \underset{a\in\mathscr{A}, |c(a)|>-\log_2\varepsilon}{\operatorname{argmin}} |c(a)|$$

Considérons alors la spécification de longueur $l_1 : \mathscr{A} \to \mathbb{N}^\star$ définie par :

$$\forall a \in \mathscr{A}, l_1(a) = \begin{cases} |c(a)| - 1 & \text{si } a = a_1 \\ |c(a)| & \text{sinon} \end{cases}$$

On peut alors vérifier que la spécification de longueur l_1 vérifie bien l'inégalité de KRAFT :

$$\sum_{a\in\mathscr{A}} 2^{-l_1(a)} = 2^{-|c(a_1)|} + \sum_{a\in\mathscr{A}} 2^{-|c(a)|} \leq \varepsilon + \sum_{a\in\mathscr{A}} 2^{-|c(a)|} \leq 1$$

D'après la réciproque de l'inégalité de KRAFT, il est donc possible de définir un code c_1 conforme à la spécification de longueur l_1, code auquel est associé une perte ε_1 définie comme ci-dessus.

Si le code c_1 obtenu n'est pas complet, on peut alors réitérer cette opération pour ce code. En effet, puisque pour tout code c la perte ε associée s'écrit :

$$\varepsilon = \sum_{i=1}^{\max_{a\in\mathscr{A}}|c(a)|} 2^{-i} - \sum_{a\in\mathscr{A}} 2^{-|c(a)|}$$

ε admet par construction une décomposition en base 2 dont le plus petit terme est minoré par $2^{-\max_{a\in\mathscr{A}}|c(a)|}$. Donc si ε n'est pas nul, il permet toujours de raccourcir le plus long code $\max_{a\in\mathscr{A}}|c(a)|$. Ainsi, la suite ε_k des pertes associées aux codes c_k est strictement décroissante tant qu'elle est non-nulle, et prend ses valeurs dans un ensemble fini : elle atteint donc 0. □

Le code c_{min} obtenu à l'issue du processus de raccourcissement est appelé le *complété* du code c. Il existe en général de nombreuses manières de compléter un code incomplet, mais la méthode présentée précédemment est non seulement générique (elle fonctionne quelque soit le code c envisagé), mais de plus assure l'ordre des lettres en terme de leur longueur de code : si a et a' sont telles que $|c(a)| < |c(a')|$, alors $|c_{min}(a)| \leq |c_{min}(a'|$.

Compte-tenu de leur propriété de minimalité de la longueur de code, les codes complets sont à privilégier dans les problèmes de compression de texte, que nous abordons à présent.

2. Compression de texte

Minimiser le nombre de bits requis pour représenter une information est essentiel : dans le cas d'une transmission télégraphique, la durée de la transmission est égale au nombre de bits à transmettre ; dans le cas d'un stockage sur un support physique, le nombre de bits disponibles est limité, et ne pas compresser l'information conduit à une exploitation sous-optimale des capacités de stockage.

On considère à présent que l'on transmet ou stocke un message $x \in \mathscr{A}^n$ composé de n symboles (n étant fixé pour simplifier le propos), et que, ayant déjà transmis un certain nombre de messages $x_1, \ldots, x_p$, on dispose d'une connaissance a priori du prochain message à transmettre, résumée par une distribution $\mathbb{P}$ sur l'espace des messages $\mathscr{A}^n$. Considérons le cas le plus général : on envisage un code c pour l'espace des messages ^{n} tout entier.

2.1. Longueur des messages. Choisir un code qui minimise la longueur du message codé étant donné une connaissance a priori du message requiert d'abord de définir en quel sens cette minimisation a lieu. On définit pour cela le taux de compression attendu d'un code sous une distribution $\mathbb{P}$.

DÉFINITION 14. *La longueur attendue d'un message de longueur n généré selon la distribution* $\mathbb{P}$ *et codé par le code c est définie comme* $\mathbb{E}_{\mathbb{P}}(|c|)$.

Bien entendu, certains messages ont une longueur attendue du code qui excède la longueur moyenne attendue, alors que d'autres ont une longueur moyenne inférieure à l'attendue.

2.2. Entropie et compression. La longueur attendue d'un message lie étroitement le code avec la distribution génératrice du message. Etant donnée une distribution $\mathbb{P}$ sur l'alphabet $\mathscr{A}$ générant le message à coder, on peut en effet rechercher le code préfixe qui minimise la longueur de code attendue.

DÉFINITION 15. *On appelle* code optimal pour la distribution $\mathbb{P}$ *le code* $c^\star(\mathbb{P})$ *pour l'espace des messages* $\mathscr{A}^n$ *tel que :*

$$\mathbb{E}_{\mathbb{P}}(|c^\star(\mathbb{P})|)) = \operatorname*{argmin}_{c \in \mathscr{C}(\mathscr{A}^n)} \mathbb{E}_{\mathbb{P}}(|c|)$$

SHANNON, en 1948, a énoncé le théorème fondamental qui, tout à la fois, exhibe la borne inférieure de la longueur de code attendue, et montre par ailleurs que cette borne peut toujours être atteinte à une unité près au moins, quelque soit la distribution $\mathbb{P}$ génératrice du message.

THÉORÈME 21 (Codage entropique de SHANNON). *Soit* $\mathscr{A}$ *un ensemble fini, et* $\mathbb{P}$ *une distribution de probabilité sur* $\mathscr{A}$. *Le code optimal* $c^\star(\mathbb{P})$ *pour la loi* $\mathbb{P}$ *vérifie :*

$$\mathbb{H}(\mathbb{P}) \leq \mathbb{E}_{\mathbb{P}}(|c^\star|) \leq \mathbb{H}(\mathbb{P}) + 1$$

où $\mathbb{H}(\mathbb{P})$ *désigne l'entropie (en base 2) de la distribution* $\mathbb{P}$.

Prouvons tout d'abord la première inégalité : soit c un code préfixe pour l'alphabet $\mathscr{A}$, et $\mathbb{P}$ la distribution génératrice du message. On a :

$$\mathbb{H}(\mathbb{P}) = -\sum_{a \in \mathscr{A}} \mathbb{P}(a) \log_2 \mathbb{P}(a)$$

Par ailleurs, considérons la distribution $\mathbb{Q}_c$ sur l'alphabet $\mathscr{A}$ telle que :

$$\forall a \in \mathscr{A}, \mathbb{Q}_c(a) = \frac{2^{-|c(a)|}}{\sum_{a \in \mathscr{A}} 2^{-|c(a)|}}$$

Comme la distance de Kullback-Leibler de $\mathbb{P}$ à $\mathbb{Q}$ est positive :

$$D(\mathbb{P}||\mathbb{Q}_c) = \sum_{a \in \mathscr{A}} \mathbb{P}(a) \log_2 \frac{\mathbb{P}(a)}{\mathbb{Q}(a)} \geq 0$$

on a :

$$\mathbb{H}(\mathbb{P}) \leq -\sum_{a\in\mathscr{A}} \mathbb{P}(a)\log_2 \mathbb{Q}(a)$$

Or :

$$\sum_{a\in\mathscr{A}} \mathbb{P}(a)\log_2 \mathbb{Q}(a) = \sum_{a\in\mathscr{A}} \mathbb{P}(a)|c(a)| + \log_2 C \tag{28}$$

et $\log_2 C < 0$ d'après l'inégalité de KRAFT. D'où :

$$\mathbb{H}(\mathbb{P}) \leq \sum_{a\in\mathscr{A}} \mathbb{P}(a)|c(a)| = \mathbb{E}_{\mathbb{P}}(|c|)$$

Tout code préfixe a donc une longueur de code attendue supérieure ou égale à l'entropie de la distribution génératrice du message, et cette inégalité est donc vraie pour le code optimal.

La seconde inégalité s'appuie sur la réciproque de l'inégalité de KRAFT. En effet, étant donnée la distribution $\mathbb{P}$, considérons la fonction de longueur de code $l : a \rightarrow -E(\log_2 \mathbb{P}(a))$, où $E(x)$ désigne la partie entière du réel x au sens du plus grand entier relatif inférieur à x. Comme, pour tout $a \in \mathscr{A}$:

$$2^{-l(a)} = 2^{-E(\log_2 \mathbb{P}(a))} < 2^{-log_2 \mathbb{P}(a)}$$

la longueur de code l vérifie bien l'inégalité de KRAFT :

$$\sum_{a\in\mathscr{A}} 2^{-l(a)} \leq \sum_{a\in\mathscr{A}} \mathbb{P}(a) = 1$$

Par conséquent, il existe un code préfixe c pour l'alphabet $\mathscr{A}$ de longueur de code l. Sa longueur de code attendue s'écrit :

$$\mathbb{E}_{\mathbb{P}}(|c|) = \sum_{a\in\mathscr{A}} \mathbb{P}(a)|c(a)|$$

soit :

$$\mathbb{E}_{\mathbb{P}}(|c|) < \sum_{a\in\mathscr{A}} \mathbb{P}(a)(1 + \log_2 \mathbb{P}(a)) = \mathbb{H}(\mathbb{P}) + 1$$

□

Ce résultat essentiel en théorie de la compression renforce l'intérêt de la réciproque de l'inégalité de KRAFT : non seulement ce résultat permet de prouver que la borne inférieure sur la longueur de code attendue peut être atteinte en un certain sens, mais elle fournit aussi la méthode pour le construire.

Par ailleurs, le théorème de SHANNON renforce, si besoin était, le fondement de l'entropie pour mesurer la quantité d'information contenue dans une variable aléatoire : il s'agit du nombre attendu de bits nécessaires à l'encodage de chacune des issues possibles de la variable aléatoire. Il est utile de noter que l'entropie décrite ici est définie en base 2, conformément au caractère binaire du codage envisagé. S'il s'agissait de réaliser un codage par des éléments pris parmi $p > 2$ (au lieu de $|\{0,1\}| = 2$ pour un code binaire), on retrouverait la même relation en définissant l'entropie en base $|p|$. La constante de BOLTZMANN k_B, qui lie l'entropie thermodynamique et l'entropie informationnelle par la relation :

$$S(\langle \boldsymbol{E} \rangle) = k_B \max_{\mathbb{P}\in\mathscr{D}(\langle \boldsymbol{E} \rangle)} \mathbb{H}(\mathbb{P})$$

apparaît donc, de ce point de vue, comme le logarithme de la *base* dans laquelle l'entropie thermodynamique est exprimée.

3. Loi de codage

3.1. Définition. Pour formuler le théorème de SHANNON, nous avons été amené à considérer parallèlement deux types de distribution de probabilité : la distribution génératrice du message, $\mathbb{P}$, que nous avons jusqu'ici considérée comme un paramètre, donc connue ; et la distribution notée $\mathbb{Q}_c$ ci-dessus, définie pour un code c, et que l'on appelle *loi de codage*. Le lien entre code et loi de codage dérive du théorème de SHANNON, qui montre que le code c est optimal pour sa loi de codage $\mathbb{Q}_c$.

DÉFINITION 16. *On appelle* loi de codage *du code c pour l'alphabet $\mathscr{A}$ la distribution $\mathbb{Q}_c$ sur $\mathscr{A}$ telle que :*

$$\forall a \in \mathscr{A}, \mathbb{Q}(a) = \frac{2^{-|c(a)|}}{\sum_{a\in\mathscr{A}} 2^{-|c(a)|}}$$

3.2. Propriétés. L'intérêt de cette notion de loi de codage réside dans les propriétés de longueur du code associé que l'on peut en dériver. En effet, l'équation (28) de la preuve du théorème de SHANNON montre que :

$$\mathbb{E}_{\mathbb{P}}(|c|) \geq \mathbb{H}(\mathbb{P}) + D(\mathbb{P}||\mathbb{Q}_c) - \log_2 \sum_{a\in\mathscr{A}} 2^{-|c(a)|}$$

La loi de codage $\mathbb{Q}_c$ apparaît donc comme le déterminant de la borne inférieure à la longueur moyenne attendue d'un message codé par le code c et généré selon la distribution $\mathbb{P}$, et ce au travers de deux termes : le logarithme de la fonction de partition de $\mathbb{Q}_c$, et la distance de KULLBACK-LEIBLER de $\mathbb{P}$ à $\mathbb{Q}_c$. Remarquons que ces deux quantités ne sont pas a priori liées, puisque la distance de KULLBACK-LEIBLER ne dépend que de la distribution normalisée.

3.2.1. *Codes complets.* Afin de minimiser la longueur moyenne attendue du message codé, on doit donc en particulier minimiser la quantité $\log_2 \sum_{a\in\mathscr{A}} 2^{-|c(a)|}$. Compte-tenu de l'inégalité de KRAFT, cette quantité est nécessairement positive. Idéalement, on peut donc annuler ce terme en choisissant un code *complet*.

A nouveau, la représentation en terme de dichotomie récursive de l'intervalle $[0,1[$ est utile pour appréhender cette notion. Chacun des intervalles $s(c(a))$ a en effet pour longueur $2^{-|c(a)|}$, et la réunion de ces intervalles a donc pour longueur $\sum_{a\in\mathscr{A}} 2^{-|c(a)|}$. Par conséquent, un code complet est associé à un ensemble d'intervalles de $[0,1[$ $\{s(c(a)), a \in \mathscr{A}\}$ qui forment une partition de $[0,1[$ tout entier. Par opposition, un code qui ne serait pas complet laisserait une partie de l'intervalle $[0,1[$ inoccupée. En d'autres termes, certains intervalles $s(c(a))$ pourraient être plus grands, et donc associés à des codes $c(a)$ plus courts.

Etant donné une fonction de longueur de code l vérifiant l'inégalité de KRAFT, il est toujours possible de trouver une fonction de longueur de code l' associée à un code complet qui soit uniformément plus courte :

$$\forall a \in \mathscr{A}, l'(a) < l(a)$$

Il suffit pour cela de coder le symbole a de code $l(a)$ le plus long par le code $c'(a)$ associé à l'intervalle $[\sum_{a'\in\mathscr{A}, a\neq a'}, 1[$.

3.2.2. *Adéquation entre loi de codage et loi génératrice.* Le second terme par lequel la loi de codage oppose une limite à la longueur moyenne attendue du message codé est la distance de KULLBACK-LEIBLER $D(\mathbb{P}||\mathbb{Q}_c)$ de la distribution génératrice $\mathbb{P}$ à la loi de codage $\mathbb{Q}_c$. Cette limite vaut pour tous les codes partageant la même loi de codage.

THÉORÈME 22. *Pour tout code c de loi de codage $\mathbb{Q}$, et toute distribution génératrice du message à coder $\mathbb{P}$, la longueur moyenne attendue du message codé vérifie :*

$$\mathbb{E}_{\mathbb{P}}(|c|) \geq \mathbb{H}(\mathbb{P}) + D(\mathbb{P}||\mathbb{Q})$$

avec égalité si et seulement si le code est complet.

La preuve de ce théorème résulte immédiatement de l'équation (28), de l'inégalité de KRAFT, ainsi que de la définition d'un code complet pour le dernier point. □

CHAPITRE 5

Statistique et compression

Jusqu'à présent, nous avons considéré la distribution génératrice du message $\mathbb{P}$ comme connue, et que sous cette distribution, les symboles du message étaient indépendants et identiquement distribués. Nous nous sommes par ailleurs cantonnés à des codes séquentiels et préfixes. Si l'on souhaite encoder un texte généré selon une distribution vérifiant ces propriétés et connue des deux utilisateurs (l'émetteur et le reveceur), on peut se contenter d'une telle situation. Mais s'il s'agit de fournir un outil informatique destiné à compresser tout type de texte, indépendamment de sa langue en particulier, il n'y a plus de connaissance a priori de la distribution génératrice du message. Il en va de même si l'on recherche une compression optimale d'une séquence génomique ou protéique sans a priori quant à l'organisme dont elle provient. Deux stratégies peuvent alors être envisagées :

- choisir une distribution de codage à partir du message particulier à transmettre, puis transmettre cette distribution avant le message. Ainsi, une convention de choix du code en fonction de la distribution estimée suffit à s'assurer que le code est adapté, d'une part, et qu'il est bien partagé par l'émetteur et le receveur, d'autre part. Mais le contrepartie est de devoir transmettre la distribution préalablement, ce qui peut avoir un coût si le message à transmettre est court.
- choisir une convention pour estimer la distribution au fur et à mesure que les symboles sont décodés, distribution à partir de laquelle le code pour la lettre suivante sera construit. Ainsi, aucune distribution n'est transmise : il suffit d'une convention pour transmettre les premiers symboles, qui peut être un code arbitraire. De telles méthodes sont dites *adaptatives*.

1. Retour au maximum d'entropie

Chacune de ces stratégies revient à assigner une distribution de probabilités étant donné l'observation d'un échantillon, constitué en l'occurrence par le message à transmettre. Elles ne diffèrent que par le nombre de fois qu'une telle assignation est effectuée, et par le fait que le résultat en soit *transmis* ou *reproduit* par le receveur. Il est tentant de dresser un parallèle avec l'approche développée dans la première partie pour assigner une distribution de probabilité à un système dont on observe une réalisation.

1.1. La longueur de code comme une énergie. Pour formuler un problème de maximum d'entropie dans le cas de la compression, il est nécessaire de définir l'*énergie* d'une particule. En l'occurrence, une *particule* est un symbole du message à compresser, et vit dans un espace d'état défini par l'alphabet $\mathscr{A}$.

Au travers du théorème de SHANNON, il apparaît que l'élément qui contraint l'entropie de la distribution génératrice du message est la longueur moyenne attendue du message codé. Il est donc naturel de considérer que l'énergie d'un symbole $a \in \mathscr{A}$ est la longueur du code qui lui est associé, $\ell(a)$.

Si l'on reformule en base 2 les quantités introduites dans la première partie, la fonction de partition associée au code c est définie par :

$$\forall \lambda \in \mathbb{R}, \quad Z_n(\lambda) = \sum_{\boldsymbol{x} \in \mathscr{A}^n} 2^{-\lambda |c(\boldsymbol{x})|} = \sum_{\boldsymbol{x} \in \mathscr{A}^n} 2^{-\sum_{a \in \mathscr{A}} \lambda N_a(\boldsymbol{x}) |c(\boldsymbol{a})|}$$

où $N_a(\boldsymbol{x})$ désigne le nombre d'occurences du symbole a dans le message $\boldsymbol{x}$.

Il est intéresssant de revenir à la représentation du code par un ensemble d'intervalles obtenus par dichotomie récursive de l'intervalle $[0,1[$, pour remarquer que la fonction de partition associée coïncide avec la somme des longueurs des intervalles $\{s(c(a)), a \in \mathscr{A}\}$. En ces termes, la fonction de partition prend tout son sens : il s'agit de la taille de l'intervalle que l'on peut partitionner en intervalles de longueur égale à chacun de ses termes. Si l'on considère la valeur de la fonction de partition pour $\lambda = 1$, $Z_1(1)$, on reconnaît le terme majoré par 1 dans l'inégalité de KRAFT. Nous verrons ci-dessous une justification de cette valeur particulière dérivée du théorème de SHANNON.

1.2. Formulation du problème de maximum d'entropie. On considère donc que l'observable qui contraint l'entropie maximale de la distribution sur les symboles est la longueur moyenne attendue d'un message codé de longueur n. Puisque l'entropie mesure la diversité des états le long du message, elle quantifie l'information contenue dans le message : de ce point de vue, toute méthode de compression cherche naturellement à maximiser la quantité d'information transmise dans le message compressé sous contrainte d'un *budget* de bits, autrement de la longueur du code.

Il est donc naturel de considérer le problème de maximum d'entropie suivant :

$$\max \mathbb{H}(\mathbb{P}) \quad s.c. \begin{cases} \sum_{a\in\mathscr{A}} \mathbb{P}(a) = 1 \\ \sum_{a\in\mathscr{A}} \mathbb{P}(a)|c(a)| \leq l \end{cases} \tag{29}$$

Mais ce problème, c'est précisément le théorème de SHANNON qui en apporte la solution. La distribution d'entropie maximale, la longueur moyenne de code attendue étant fixée, est d'après le théorème de SHANNON la loi de codage associée au code c. Mieux, le théorème de SHANNON nous montre directement que :

$$\max \mathbb{H}(\mathbb{P}) = l$$

Autrement dit, longueur moyenne attendue d'un code et entropie de la distribution de codage associée sont une seule et même quantité. Ce qui explique que la valeur du multiplicateur de LAGRANGE associé à cette contrainte, $\lambda = 1$, qui apparaît dans l'inégalité de KRAFT, soit privilégiée : un gain d'un bit sur la longueur de code se traduit par le gain d'un bit d'entropie maximal d'un message transmis.

Enfin, naturellement, on retrouve que la distribution de maximum d'entropie $\mathbb{P}^\star = \operatorname{argmax}\mathbb{H}(\mathbb{P})$ vérifie :

$$\forall a \in \mathscr{A}, \mathbb{P}^\star(a) = \frac{1}{Z(1)} 2^{-|c(a)|}$$

Autrement formulé, la loi de codage $\mathbb{Q}_c$ associé au code c est la distribution de maximum d'entropie sous contrainte de la longueur moyenne attendue du message compressé.

1.3. Choix de la distribution de codage. Explorons à présent le cas d'un message $\boldsymbol{x} \in \mathscr{A}^\star$ à transmettre, pour lequel on dispose d'une connaissance a priori issue de messages préalablement transmis, ou encore d'un corpus de texte supposé représentatif de l'usage des symboles dans le message à transmettre. Cette connaissance a priori peut être synthétisée sous la forme de statistiques, par exemple en dénombrant les mots de p lettres. On dispose alors d'une connaissance a priori du message sous la forme de comptages des mots de p lettres, notée $\langle \boldsymbol{N} \rangle = (\langle N_{\boldsymbol{w}} \rangle)_{\boldsymbol{w}\in\mathscr{A}^p}$. Remarquons ici que l'on pourrait tout aussi bien utiliser n'importe quelle statistique, mais lorsqu'il s'agit de textes en langages naturels, il est attendu que les biais de composition soient locaux, et par conséquent correctement capturés par les comptages des mots.

Il s'agit à présent de déterminer une distribution sur le message à transmettre qui, sous l'hypothèse que ce dernier soit statistiquement similaire à la connaissance a priori,

permette de minimiser la longueur du message codé. Compte-tenu du théorème de SHANNON, on sait dores et déjà que la longueur optimale attendue du message codé est bornée inférieurement par l'entropie de cette distribution.

Considérer la connaissance a priori sur le message conduit à considérer des distributions de probabilité sur le message appartenant à l'ensemble $\mathscr{D}(\langle \boldsymbol{N} \rangle)$ des distributions prédisant les statistiques observées $\langle \boldsymbol{N} \rangle$, au sens où :

$$\forall \mathbb{P} \in \mathscr{D}, \mathbb{E}_{\mathbb{P}}(\boldsymbol{N}) = \langle \boldsymbol{N} \rangle$$

Mais, en général, il existe un grand nombre de telles distributions de probabilité. Chacune d'elle impose, au travers du théorème de SHANNON, la longueur optimale attendue du message codé (le qualificatif *attendue* s'entend ici *compte-tenu de la connaissance a priori considérée*). Considérer la distribution $\mathbb{P}$ sur $\mathscr{A}^n$, où n désigne la longueur du message $\boldsymbol{x} \in \mathscr{A}^n$ à transmettre, conduit en effet à la relation :

$$\forall c \in \mathscr{C}(\mathscr{A}^n), \mathbb{H}(\mathbb{P}) \leq \mathbb{E}_{\mathbb{P}}(|c(\boldsymbol{x})|$$

Cette borne inférieure peut par ailleurs toujours être atteinte à un bit près, moyennant le choix du code associé à la loi de codage la plus proche, au sens de la distance de KULLBACK-LEIBLER, de la distribution supposée $\mathbb{P}$ du message.

1.3.1. *Retour au maximum d'entropie.* Ainsi, le principe de maximum d'entropie trouve ici une nouvelle justification : compte-tenu des remarques précédentes, déduire le code optimal d'une distribution qui ne maximise pas l'entropie sur l'ensemble $\mathscr{D}(\langle \boldsymbol{N} \rangle)$ conduit inévitablement à s'attendre à une longueur du message codé d'autant plus faible. Le principe de maximum d'entropie apparaît donc ici comme un principe *min-max*, au sens où il dicte de choisir, parmi les distributions compatibles avec la connaissance a priori, celle qui maximise la longueur du message codé par le code optimal sous cette distribution. Ne pas appliquer ce principe conduirait à s'attendre à une longueur de code inférieure à celle réellement atteignable. Pire, compte-tenu du phénomène de concentration de l'entropie, ce risque de sous-estimation de la longueur attendue du message codé serait croissant avec la longueur du message.

1.3.2. *Justification des codes séquentiels.* Dans le cas envisagé ici, où la connaissance a priori sur le message est synthétisée sous la forme d'une statistique représentant les comptages des mots de p lettres dans l'ensemble d'apprentissage, nous savons dores et déjà que la distribution maximisant l'entropie est une distribution d'indépendance lorsque $p = 1$, et une distribution de MARKOV d'ordre $p-1$ lorsque $p > 1$.

Ce point est d'une importance primordiale pour la compression, comme nous allons le mettre en évidence dans le cas $p = 1$, c'est-à-dire dans le cas d'une distribution d'entropie maximale sous laquelle les symboles du message sont indépendants. Dans ce cas en effet, si l'on note $\mathbb{P}_n^\star$ la distribution de maximum d'entropie compatible avec la connaissance a priori de la composition en symboles du message, on sait qu'il existe une distribution $\mathbb{P}^\star$ sur l'alphabet telle que :

$$\forall \boldsymbol{x} \in \mathscr{A}^n, \mathbb{P}_n^\star(\boldsymbol{x}) = \mathbb{P}^{\star \otimes n}(\boldsymbol{x}) \tag{30}$$

Il est immédiat de remarque que, dans ce cas, l'entropie $\mathbb{H}(\mathbb{P}_n^\star)$ se déduit de l'entropie de la distribution *instantanée* :

$$\mathbb{H}(\mathbb{P}_n^\star) = n \times \mathbb{H}(\mathbb{P}^\star)$$

Par conséquent, la borne de SHANNON sur la longueur attendue optimale du message codé s'écrit :

$$n \times \mathbb{H}(\mathbb{P}^\star) \leq \mathbb{E}_{\mathbb{P}^{\star \otimes n}}(|c(\boldsymbol{x})|)$$

soit encore :

$$\mathbb{H}(\mathbb{P}^\star) \leq \frac{1}{n} \mathbb{E}_{\mathbb{P}^{\star \otimes n}}(|c(\boldsymbol{x})|)$$

Or, si le code c est choisi séquentiel, la longueur de la séquence codée est la somme des longueurs des codes des symboles, soit :

$$\mathbb{H}(\mathbb{P}^\star) \leq \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathbb{P}^\star}(|c(x_i)|)$$

Les arguments de la somme dans le membre de droite ne dépendant pas de la position dans la séquence (puisque l'on prend l'espérance contre une distribution invariante le long de la séquence), on peut encore réécrire la relation précédente :

$$\mathbb{H}(\mathbb{P}^\star) \leq \mathbb{E}_{\mathbb{P}^\star}(|c(x)|)$$

Ceci établit le théorème suivant :

THÉORÈME 23. *Lorsque la connaissance a priori sur le message à transmettre est la statistique du nombre d'occurrences de chacun des symboles, la longueur attendue optimale du message codé est atteinte pour le code séquentiel optimal pour la distribution* $\mathbb{P}^\star$ *sur l'alphabet* $\mathscr{A}$ *telle que* $\mathbb{P}^\star_n = \mathbb{P}^{\star\otimes n}$.

Ce résultat peut naturellement être généralisé au cas où les statistiques résumant la connaissance a priori ne sont pas les comptages des symboles, mais ceux des mots d'une longueur p fixée, autrement dans le cas où la distribution d'entropie maximale est une chaîne de MARKOV.

2. Codages adaptatifs

Nous avons vu au chapitre précédent comment utiliser une connaissance a priori des messages à transmettre pour choisir un code permettant une compression optimale du message. Cela nous a permis de justifier le recours au maximum d'entropie sous la forme d'un critère min-max, le choix de la distribution de maximum d'entropie revenant à maximiser la longueur du code optimal associé parmi les distributions prédisant les comptages des symboles composant le message.

Nous envisageons à présent le cas de l'absence de connaissance a priori de la composition en symboles des messages à transmettre. Compte-tenu du lien étroit entre codes et distributions de probabilité exposé précédemment, il reste nécessaire de choisir une distribution de codage. Une approche consiste à utiliser les symboles déjà transmis du message en lieu et place de la connaissance a priori utilisée précédemment pour choisir la distribution de codage en chaque position.

Cette généralisation des méthodes de compression soulève quelques problèmes spécifiques. Le premier de ceux-ci concerne les premiers symboles de la séquence : pour le tout premier, aucun symbole n'a encore été transmis, et aucune information n'est par conséquent disponible pour conduire le choix de cette distribution ; pour au moins quelques symboles suivants, une information très partielle sera disponible, et peut conduire aux choix de distributions de codage peu efficaces. Nous verrons dans cette section que le principe de maximum d'entropie justifie le recours aux méthodes bayésiennes pour réaliser ces choix de distribution de codage.

Le second problème soulevé par ce changement d'approche dérive de la dépendance de la distribution de codage en fonction de la position dans la séquence. Chaque symbole, lorsqu'il est transmis, alimente en effet l'échantillon exploité pour le choix de la distribution de codage. Ainsi, pour chaque position, la distribution de codage utilisée peut varier (et varie en pratique, surtout au début de la séquence où le poids relatif d'un nouveau symbole n'est pas négligeable). Ceci impose que l'émetteur et le receveur s'accordent sur la manière d'utiliser les informations contenues dans la partie déjà transmise du message pour décider de la distribution de codage de chaque symbole.

Enfin, un phénomène lié au recours à l'approche bayésienne surprend au premier abord : même si les distributions de codage choisies en chaque position ne prennent pas en compte de dépendance vis-à-vis du passé de la séquence, le fait que la distribution

elle-même soit *apprise* à partir du passé de la séquence induit une telle dépendance. Et dans le cas plus général de distributions de codage présentant des dépendances entre les positions, comme les chaînes de Markov par exemple, ce phénomène restera présent et induira des dépendances typiquement plus longues que l'ordre des chaînes de Markov envisagées.

Nous resterons, dans cette section, dans le cadre de distributions de codage sans dépendance, autrement dit nous n'exploiterons, pour choisir les distributions de codage en une position donnée, que les comptages des symboles déjà transmis.

2.1. Du maximum d'entropie à la statistique bayésienne.

2.1.1. *Démarche.* Laissons provisoirement de coté le problème du choix de la distribution de codage pour le premier symbole de la séquence, pour se concentrer sur la question du choix d'une distribution de codage pour une position particulière de la séquence.

Soit donc un message $\boldsymbol{x} \in \mathscr{A}^n$, dont $t < n$ symboles ont déjà été transmis. Parmi ces p symboles, il est possible de calculer le nombre d'occurences $n_t(a)$ de chacun des symboles a de l'alphabet $\mathscr{A}$. Si l'on applique le même principe que précédemment pour choisir la distribution de codage utilisée pour transmettre le symbole suivant, alors on choisira la distribution $\mathbb{P}_t$ sur l'alphabet qui prédit la composition de l'ensemble d'entraînement (ici le début de la séquence), c'est-à-dire les fréquences dérivées des comptages :

$$\forall\, t \in \{1,\dots,n\}, \forall\, a \in \mathscr{A}, \mathbb{P}_t(a) = \frac{n_t(a)}{\sum_{a\in\mathscr{A}} n_t(a)} = \frac{n_t(a)}{t}$$

Mais si l'on admet que l'occurrence de chacun des symboles de l'alphabet en chacune des positions est aléatoire, alors la cohérence de la démarche impose de considérer également que les comptages $(n_t(a))_{a\in\mathscr{A}}$ auraient pu être différents. Plus précisément, il faut considérer que ces comptages sont aléatoires, et qu'ils répondent à la loi multinomiale induite par la distribution $\mathbb{P}_t$ sur une séquence de longueur t.

2.1.2. *Distributions conjuguées.* Mais puisque ces comptages sont aléatoires, alors la distribution $\mathbb{P}_t$ estimée en les utilisant l'est également. Il s'agit donc d'une nécessité de cohérence de considérer cette distribution elle-même comme aléatoire. Or, il s'agit précisément de la différence fondamentale entre statistique classique et bayésienne. Plutôt qu'un choix, le recours à la statistique bayésienne apparaît donc comme une nécessité[1].

En l'occurence, le paramètre $(\mathbb{P}_t(a))$ de la distribution sur l'alphabet pour la position $t+1$ du message suit une distribution multinomiale dûment normalisée, de manière à ne charger que le simplexe $S_{|\mathscr{A}|} = \{\boldsymbol{x} \in [0,1]^{|\mathscr{A}|}, \sum_{a\in\mathscr{A}} x_a = 1\}$.

THÉORÈME 24. *Sous la distribution prédisant les comptages des symboles observés sur les t premiers symboles du message, les paramètres $(\theta_a)_{a\in\mathscr{A}} = (\mathbb{P}_t(a))_{a\in\mathscr{A}}$ de la distribution estimée sur ces comptages suivent la distribution de probabilité $\mathbb{Q}_t$ sur le simplexe, définie par sa densité q_t par rapport à la mesure de* LEBESGUE *:*

$$\forall (p_a)_{a\in\mathscr{A}} \in S_{|\mathscr{A}|}, q_t((p_a)_{a\in\mathscr{A}}) = \frac{1}{B(\boldsymbol{n}_t)} \prod_{a\in\mathscr{A}} p_a^{n_t(a)-1}$$

où $B(\boldsymbol{n}_t)$ désigne la fonction beta multinomiale*, dont la définition est :*

$$\forall \boldsymbol{\alpha} \in S_{|\mathscr{A}|}, B(\boldsymbol{\alpha}) = \frac{\prod_{a\in\mathscr{A}} \Gamma(\alpha_a)}{\Gamma(\sum_{a\in\mathscr{A}} \alpha_a)}$$

La distribution q_t est connue comme la distribution de DIRICHLET *de paramètre $\boldsymbol{n}_t$, notée par la suite $\mathscr{D}(\boldsymbol{n}_t)$.*

---

[1] Ce point constitue l'argument principal d'une réponse (voir [**?**]) faite à l'auteur de [**?**], attaquant le titre de ce livre, *le choix bayésien*. Il y est affirmé, au jour de l'approche *jaynésienne* des statistiques, qu'*il n'y a pas de choix bayésien.*

La preuve de ce théorème peut être trouvée dans [**?**].

La distribution q_t sur les distributions de probabilité sur l'alphabet est appelée *distribution conjuguée* de la distribution de maximum d'entropie sur la moyenne observée. Ici, la distribution de DIRICHLET apparaît donc comme la distribution conjuguée de la distribution multinomiale.

2.1.3. *Mise à jour de la distribution sur les paramètres.* Remarquons que, contrairement au principe de maximum d'entropie qui a émergé des développements de la physique statistique il y a environ 150 ans, la nécessité du recours à la statistique bayésienne n'apparaît pas dans ces approches. Cela s'explique par le fait que les échantillons de matière considérés par les physiciens sont constitués d'un tellement grand nombre de particules, que de toute façon les moyennes observées sur ses échantillons n'ont plus aucun caractère aléatoire, ayant convergé vers leurs *vraies* valeurs : la physique statistique n'a donc, historiquement, pas été confrontée à ce type de situation dans le processus d'élaboration des fondements statistiques de la thermodynamique. La situation particulière du codage *en ligne* tel que nous l'envisageons ici se prête donc particulièrement bien à l'élaboration des fondements de la statistique bayésienne comme une conséquence du principe de maximum d'entropie justifiée par un impératif de cohérence.

Explorons à présent la mise à jour de la distribution sur le paramètres lorsqu'un nouveau symbole est observé. Pour ce faire, reproduisons à présent le même raisonnement pour le symbole en position $t+1$ de la séquence : la distribution q_{t+1} est, de manière similaire, une distribution de DIRICHLET. En revanche, les paramètres en sont différents, puisque $q_{t+1} \sim \mathscr{D}(\boldsymbol{n}_{t+1})$. Quelles relations entretiennent q_t et q_{t+1} ?

Pour répondre à cette question, il faut remarquer que, x_{t+1} étant fixé :

$$\forall \boldsymbol{\theta} \in S_{|\mathscr{A}|}, \mathbb{P}(x_{t+1}|\boldsymbol{\theta}) \times q_t(\boldsymbol{\theta}) = q_t(\boldsymbol{\theta}) \times \theta_{x_{t+1}}$$

soit encore :

$$\mathbb{P}(x_{t+1}|\boldsymbol{\theta}) \times q_t(\boldsymbol{\theta}) = \frac{1}{B(\boldsymbol{n}_t)} \prod_{a \in \mathscr{A}} \theta_a^{n_t(a)-1} \theta_{x_{t+1}} = \frac{1}{B(\boldsymbol{n}_t)} \prod_{a \in \mathscr{A}} \theta_a^{n_{t+1}(a)-1}$$

Or, si x_{t+1} est fixé, cette densité de probabilité sur le simplexe est identique, à la normalisation près, à la densité q_{t+1}. Mais cela suffit à établir le résultat suivant.

THÉORÈME 25. *Soit $\boldsymbol{x} \in \mathscr{A}^n$ un message, et soit $t \in \{1, \dots, n\}$. On note $\boldsymbol{n}_t = (n_t(a))_{a \in \mathscr{A}}$ les comptages de chacun des symboles $a \in \mathscr{A}$ dans la sous-séquence $\boldsymbol{x}_{1,t}$, et q_t la distribution conjuguée de la loi multinomiale prédisant les comptages $\boldsymbol{n}_t$. Alors :*

$$\forall \boldsymbol{\theta} \in S_{|\mathscr{A}|}, q_{t+1}(\theta) = q_t(\theta|x_{t+1})$$

Autrement dit, le schème bayésien consistant à définir une loi a priori sur les paramètres, puis calculer la distribution a posteriori conditionnellement à une observation de la variable modélisée revient à *mettre à jour* la distribution sur le paramètre lorsqu'une nouvelle observation est acquise. En ce sens, la distribution courante (*i.e.* après acquisition d'un certain nombre d'observations) sur le paramètre résume entièrement l'information contenue dans l'échantillon, et se substitue sans perte à l'échantillon lui-même.

2.1.4. *Choix de la distribution a priori.* Il reste cependant une situation où le point de vue développé précédemment n'apporte pas de réponse définitive : il s'agit du choix de la distribution utilisée pour encoder le premier symbole du message. Dans ce cas, la distribution a priori est définie sans recours à une information préalablement disponible.

Historiquement, diverses distributions a priori ont été proposées. Elle correspondent toutes à des approches différentes, et présentent chacune des propriétés spécifiques qui peuvent les justifier, ou, au contraire, en restreindre la validité. Puisqu'elles correspondent à une absence totale d'information sur le paramètre de la distribution (qui,

rappelons-le, résume l'information acquise par l'observation, qui, en l'occurrence, est néante), elles sont justifiées par différentes manières de définir une distribution *non-informative*.

Distribution a priori de LAPLACE. Cet objectif assigné d'utiliser une distribution a priori qui ne porte aucune information (qui serait alors arbitraire) conduit intuitivement à considérer la distribution uniforme comme un bon candidat : elle constitue en effet la distribution maximisant l'entropie. Cependant, le principe de maximum d'entropie ne s'applique pas aussi simplement que lorsqu'il s'agit d'assigner une distribution de probabilité sur un alphabet fini. En effet, dans le cas d'un espace à probabiliser continu, la définition de l'entropie est sensible au paramétrage de l'espace.

Distribution a priori de JEFFREYS. Proposé par Harold JEFFREYS dans un article de 1946 ([**?**]) comme une distribution invariante par changement du paramétrage de la distribution, le prior de JEFFREYS est défini en fonction de l'information de FISCHER des distributions du modèle. Il répond à la définition suivante, que nous formulons ici dans le cadre de notre étude, c'est-à-dire de distributions sur un alphabet fini $\mathscr{A}$.

DÉFINITION 17 (Prior de JEFFREYS). *Soit $\mathscr{D}(\mathscr{A})$ un ensemble de distributions sur l'alphabet $\mathscr{A}$ paramétré par un vecteur $\boldsymbol{\theta} \in D \subset \mathbb{R}^d$, où D est un domaine de $\mathbb{R}^d$ et d un entier strictement positif. On appelle* prior de JEFFREYS *la distribution sur D définie par sa densité q par rapport à la mesure de* LEBESGUE *:*

$$\forall \boldsymbol{\theta} \in D, q(\boldsymbol{\theta}) \sim \sqrt{I(\boldsymbol{\theta})}$$

*où $I(\boldsymbol{\theta})$ désigne l'*information de FISCHER *de la distribution $\mathbb{P}_{\boldsymbol{\theta}} \in D$, définie par la relation $I(\boldsymbol{\theta}) = -\mathbb{E}_{\boldsymbol{\theta}}\left(\frac{\partial^2 \ln \mathbb{P}_{\boldsymbol{\theta}}}{\partial^2 \boldsymbol{\theta}}\right)$.*

L'invariance de cette distribution vis-à-vis d'un changement de paramétrage du modèle provient du fait que l'information de FISCHER, vue comme application de D dans $\mathbb{R}$, est une quantité intrinsèque. Par ailleurs, le choix d'une distribution a priori sur le paramètre proportionnelle à l'information de FISCHER pour chaque distribution du modèle permet de privilégier les distributions pour lesquelles l'information de FISCHER est grande ; compte-tenu du théorème de CRAMER-RAO, qui établit que la variance/covariance du meilleur estimateur sans biais d'un modèle régulier est l'inverse de l'information de FISCHER. En d'autres termes, le prior de JEFFREYS accorde un poids supérieur aux distributions du modèle dont les limites intrinsèques à la qualité d'estimation sont les plus faibles.

Distribution a priori de Krichevsky-Trofimov. Une alternative populaire aux distributions a priori présentées précédemment a été introduite par KRICHEVSKY et TROFIMOV en 1981 dans [**?**]. Cette distribution, baptisée d'après les noms de ces deux auteurs, est une distribution de DIRICHLET de paramètres égaux à 1/2.

Introduite dans le cadre de la compression de texte, le choix de ces paramètres conduit à une distribution dont le mode est formé par les bords du simplexe $S_{|\mathscr{A}|}$. Il privilégie ainsi les distributions d'entropie faible.

Une manière d'évaluer la qualité d'une distribution a priori consiste à étudier les résultats qu'elle fournit lorsque l'on cherche à estimer l'entropie d'une séquence en les utilisant. Cette démarche a été conduite dans [**?**], et conduit à des résultats intéressants : les choix classiques de paramètres de la distribution a priori de DIRICHLET sont évalués en terme de leur inertie vis-à-vis du problème d'estimation de l'entropie. On entend ici par inertie la quantité de données nécessaires pour *dominer* l'impact de la distribution a priori. Il apparaît que les distributions de DIRICHLET en général conduisent à des résultats décevants : lorsque les paramètres de la distribution de DIRICHLET sont trop faibles, l'entropie est systématiquement sous-estimée, les symboles rares étant considérés comme absents tant qu'ils ne sont pas vus un certain nombre de fois, ce qui exige une longueur minimale pour la séquence. A l'inverse, les paramètres trop grands conduisent à une estimation de l'entropie quasiment arbitraire.

Distribution a priori de SCHURMANN-GRASSBERGER. Les auteurs de [**?**] mentionnent cependant une distribution a priori qui réalisé un compromis entre ces biais : la distribution de SCHURMANN-GRASSBERGER, pour laquelle les paramètres de la distribution de DIRICHLET sont choisis égaux à $1/|\mathscr{A}|$. Cette distribution a priori fut introduite dans [**?**] dans le cadre de l'analyse des systèmes chaotiques, pour laquelle l'estimation de l'entropie d'une séquence de symboles est un élément clé. Pour le critère retenu par les auteurs pour évaluer la qualité des distributions a priori, à savoir la variance de l'estimateur de l'entropie, cette distribution a priori apparaît comme optimale : il s'agit en effet de celle pour laquelle cette variance est la plus élevée. En d'autres termes, la distribution a priori de SCHURMANN-GRASSBERGER est celle qui, parmi les distributions de DIRICHLET, impose le moins une valeur particulière de l'entropie estimée. A l'inverse, une distribution a priori sous laquelle la variance de l'estimateur de l'entropie est faible (comme le rapportent les auteurs pour la distribution de KRICHESKY-TROFIMOV) fournit certes un estimateur très reproductible, mais en général faux : ce ne sont alors pas les données qui fournissent l'estimation, mais la distribution a priori.

2.2. Codages adaptatifs. Mais revenons à l'objectif initial de ce chapitre, à savoir l'utilisation, en chaque position du message à transmettre, des symboles déjà transmis pour définir la distribution de codage du prochain symbole à transmettre. Ayant adopté pour ce faire un point de vue bayésien, la connaissance courante sur ce symbole n'est plus décrite par une unique distribution, mais pas une distribution de distributions de probabilité. Comment en déduire une loi de codage ?

Il existe en fait une solution assez simple à ce problème. Bien que l'approche bayésienne consiste à récursivement mettre à jour la distribution sur le paramètre de la loi d'indépendance, il est en effet possible de calculer la probabilité de toute séquence sous le modèle bayésien, ce que nous verrons dans le paragraphe suivant. Nous verrons dans un second temps que cette distribution de probabilité sur les séquences permet de dériver aisément la distribution du symbole courant sous ce même modèle bayésien, et par conséquent de raccrocher au schéma de codage présenté en début de cette partie.

2.2.1. *Calcul de la probabilité d'une séquence.* Formellement, le calcul de la probabilité d'une séquence sous le modèle bayésien introduit précédemment revient à une sorte d'*intégration* le long de la séquence : chaque symbole est généré selon un mélange de distributions, ce mélange étant lui-même dépendant des symboles précédents, et il s'agit par conséquent de sommer récursivement les probabilités des symboles le long de la séquence.

Une remarque préliminaire semble importante ici : même si les symboles précédents le symbole courant du message ne sont pris en compte que sous la forme de leurs nombres d'occurences, autrement dit même si l'on se restreint à des distributions d'indépendance, le fait que l'on mélange les distributions d'indépendance par rapport à une distribution sur leurs paramètres qui est mise à jour en chaque position ôte à la distribution résultante sur les messages son caractère indépendant. En résumé, la distribution sur chaque symbole du message est certes formulée comme une distribution d'indépendance, mais ses paramètres dépendant du passé de la séquence, le symbole n'est pas indépendant de ceux qui le précèdent.

Cette propriété, déroutante de prime abord, persiste dans le cas de distributions markoviennes : même si la connaissance a priori sur chaque symbole de la séquence est formulée comme un mélange de distributions à mémoire bornée, le fait que les pondérations des différentes distributions de ce type dépende du passé de la séquence induit une dépendance statistique du symbole courant vis-à-vis de l'ensemble du début de la séquence.

Malgré cela, il reste possible d'évaluer la probabilité d'une séquence sous le modèle bayésien de manière assez simple. Soit $\boldsymbol{x} \in \mathscr{A}^n$ un message, et notons $\mathbb{P}(\boldsymbol{x})$ sa probabilité sous le modèle bayésien. Si l'on admet que la probabilité de chaque symbole s'obtient en moyennant sa probabilité obtenue pour chaque valeur du paramètre contre la distribution $q_t(\boldsymbol{\theta})$ sur le paramètre $\boldsymbol{\theta}$ en cette position, on a alors :

$$\mathbb{P}(\boldsymbol{x}) = \prod_{t=1}^{n} \int_{\boldsymbol{\theta} \in S_{|\mathscr{A}|}} \mathbb{P}(x_t|\boldsymbol{\theta}) \times q_{t-1}(\boldsymbol{\theta}) \tag{31}$$

où q_0 désigne la distribution a priori sur le paramètre, à choisir parmi les diverses distributions de DIRICHLET introduites précédemment. Cette quantité se calcule grâce au résultat suivant.

THÉORÈME 26. *La probabilité $\mathbb{P}(\boldsymbol{x})$ d'un message $\boldsymbol{x} \in \mathscr{A}^n$ sous le modèle bayésien avec la distribution a priori de Dirichlet de paramètres $\boldsymbol{\alpha}$ vérifie :*

$$\mathbb{P}(\boldsymbol{x}) = \frac{1}{B(\boldsymbol{\alpha})} \int_{\boldsymbol{\theta} \in S_{|\mathscr{A}|}} \prod_{a \in \mathscr{A}} \theta_a^{\alpha_a + N_a(\boldsymbol{x}) - 1} \tag{32}$$

La preuve de ce théorème s'appuie sur une élimination en cascade des termes le long de la séquence. En effet, si l'on note $\boldsymbol{n}^t(\boldsymbol{x}) = \left(n_a^{(t)}\right)_{a \in \mathscr{A}}$ les comptages des symboles dans la séquence extraite $\boldsymbol{x}_{1,t}$, alors :

$$\forall\, t \in \{1, \ldots, n\}, \forall\, \boldsymbol{\theta} \in S_{|\mathscr{A}|}, q_t(\theta) = \frac{1}{B(\boldsymbol{\alpha} + \boldsymbol{n}^{(t-1)})} \prod_{a \in \mathscr{A}} \theta_a^{\alpha_a + n_a^{t-1}(\boldsymbol{x}) - 1}$$

Comme :

$$B(\boldsymbol{\alpha} + \boldsymbol{n}^{(t-1)}) = \int_{\boldsymbol{\theta} \in S_{|\mathscr{A}|}} \prod_{a \in \mathscr{A}} \theta_a^{\alpha_a + n_a^{(t-1)}(\boldsymbol{x}) - 1}$$

chaque terme intégral d'indice t dans 31 se simplifie avec la constante de normalisation $B(\boldsymbol{\alpha} + n^{(t)})$. □

Grâce à la relation précédente, il est donc possible de calculer la probabilité de n'importe quelle séquence. En particulier, pour tout message $\boldsymbol{x} \in \mathscr{A}^n$, et pour tout $t \in \{1, \ldots, n\}$, il est possible d'évaluer $\mathbb{P}(\boldsymbol{x}_{1,t})$. Par ailleurs,on peut envisager toutes les manières d'ajouter un symbole $a \in \mathscr{A}$ à la séquence $\boldsymbol{x}_{1,t}$, et calculer pour chacun de ces cas $\mathbb{P}(\boldsymbol{x}_{1,t}a)$. On peut alors en déduire la probabilité que le symbole x_{t+1} soit a, pour tout $a \in \mathscr{A}$:

$$\mathbb{P}(x_{t+1} = a|\boldsymbol{x}_{1,t}) = \frac{\mathbb{P}(\boldsymbol{x}_{1,t}a)}{\mathbb{P}(\boldsymbol{x}_{1,t})}$$

Dès lors, on peut rechercher le code $c_{t+1} \in \mathscr{C}(\mathscr{A})$ optimal pour cette distribution, et encoder le $t+1^{\grave{e}}$ symbole avec ce code en recourant à n'importe quelle méthode permettant de construire un code optimal pour cette distribution. Cette approche, mise en œuvre dans divers algorithmes de codage, permet à l'émetteur et au receveur du message de déduire la même distribution pour le symbole suivant (puisque les symboles déjà transmis sont bien entendu connus d'eux deux), et donc le même code moyennant qu'ils partagent le même algorithme de choix d'un code optimal.

Enfin, il est utile de regrouper ici quelques remarques quant à la distribution sur les séquences induite par le modèle bayésien. Comme mentionné précédemment, bien que fondée sur des observations d'occurrences de symboles le long de la séquence, et donc sur un modèle d'indépendance des symboles du message, ces derniers ne sont pas indépendants sous le modèle bayésien : l'ensemble des symboles précédant une position de la séquence participent en effet à façonner la distribution sur le symbole en cette position. Un autre point de vue sur cette propriété consiste à s'interroger sur le sens de cette distribution de probabilité : il s'agit finalement du cumul, le long de la séquence, des probabilités telles qu'attendues compte-tenu du début de la séquence, d'observer les symboles que l'on observe. En ces termes, la probabilité d'une séquence

sous le modèle bayésien est d'autant plus grande que celle-ci reproduit, de manière homogène, les mêmes propriétés statistiques.

3. Arbres de contexte

Nous avons jusqu'à présent considéré principalement le cas où la connaissance a priori sur la composition en symboles de la séquence était résumée par la statistique des occurrences de chacun des symboles dans l'ensemble d'apprentissage, restreignant par là-même la classe de modèles étudiée à des modèles d'indépendance. Nous présentons dans le suite une famille de méthodes permettant de prendre en compte des interactions entre les symboles voisins.

S'il est possible de prendre en compte ces interactions en choisissant une longueur p des mots dénombrés pour résumer le contenu de la séquence en une statistique, et donc en considérant la classe de modèles des chaînes de Markov d'ordre $p-1$, il est utile cependant de disposer de critères permettant le choix de *bonnes* valeurs de p compte-tenu du contenu de la séquence. Ce problème, qui relève du domaine de la sélection de modèles, a été abordé de diverses manières, par exemple en recourant à l'optimisation de critères quantifiant la qualité d'ajustement des différents modèles aux données. Il a ainsi été montré que le critère BIC, sur lequel nous reviendrons de nombreuses fois dans la suite, permettait une sélection consistante de l'ordre d'un modèle de Markov [**?**].

Nous nous intéresserons ici à des approches plus générales, qui n'imposent pas nécessairement de recourir à un ordre p de dépendances uniforme pour toutes positions de la séquence, mais plutôt permettent d'adapter cet ordre en fonction des symboles impliqués dans l'interaction. Du point de vue statistique, il s'agit de choisir une liste de mots de longueurs diverses, dont les nombres d'occurrences dans la séquence forment les observables du modèle. Cependant, une approche aussi générale est déraisonnable, du moins si l'on s'assigne pour objectif de *découvrir*, dans la ou les séquences étudiées, la liste de mots la plus à même, sous un certain critère, d'expliquer les données. Aussi, il est nécessaire de restreindre les listes de mots envisagées pour construire le modèle. En 1983 que Jorma RISSANEN publie un article intitulé *A universal data compression system* [**?**]. Cet article introduit un algorithme, baptisé *context* par l'auteur, qui propose de représenter les dépendances entre les symboles apparaissant en des positions successives de la séquence par un arbre de dépendance, dont la structure restreint précisément la liste de mots envisagés. L'algorithme proposé par l'auteur s'attache en particulier à déterminer l'arbre de dépendances qui permet la meilleure compression de la séquence.

Ce chapitre introduit la représentation arborescente des dépendances au sein d'une séquence, pour ensuite introduire les *arbres de contexte* (ou, de manière équivalente, les *fonctions de contexte*) à proprement parler. L'algorithme *context*, tel qu'introduit par RISSANEN, est ensuite présenté, puis analysé d'après les résultats de P. BÜHLMANN. Enfin, on décrira dans la section suivante deux méthodes très populaires en compression, *Context Tree Maximization* et *Context Tree Weighting*, ainsi que les principaux résultats afférents. Ce sera le lieu de l'intégration, en un seul et même objet, des arbres de contexte et des méthodes issues de la compression présentées auparavant.

3.1. Représentation arborescente des chaînes de Markov. Une manière de représenter les dépendances au sein d'une chaîne de Markov d'ordre l consiste à recourir à un arbre. Les nœuds de cet arbre sont étiquetés par les lettres de l'alphabet $\mathscr{X}$, et chaque nœud interne possède exactement un nœud fils étiqueté par chacune des lettres de l'alphabet. La figure 1 propose un exemple d'arbre de dépendance d'une chaîne de Markov d'ordre 2. Cet arbre possède donc $|\mathscr{X}|^{l+1}$ nœuds, dont $|\mathscr{X}|^l$ feuilles.

Identifions chacune des feuilles de l'arbre de dépendance avec la séquence $\boldsymbol{w} = (w_1, \ldots, w_l)$ de lettres rencontrées en remontant depuis la feuille jusqu'à la racine. Une

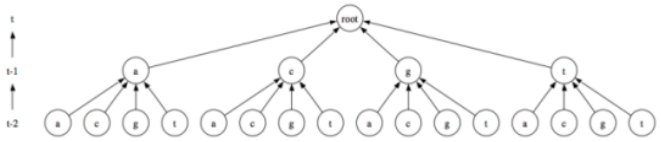


FIG. 1. Exemple d'un arbre de dépendance d'une chaîne de Markov d'ordre 2 sur l'alphabet des nucléotides $\mathscr{X} = \{a, c, g, t\}$.

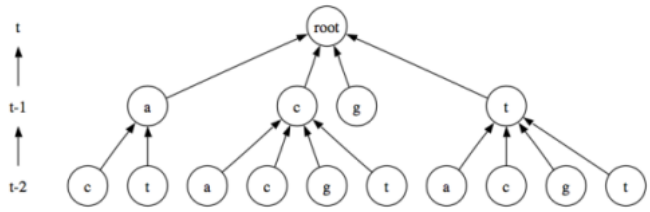


FIG. 2. Exemple d'arbre de dépendence d'ordre maximal 2, sur un alphabet de nucléotides $\mathscr{X} = \{a, c, g, t\}$.

feuille de l'arbre représente alors un *contexte* particulier $\boldsymbol{w}$ de la chaîne de Markov, auquel est associée une distribution conditionnelle $\left(\mathbb{P}(X_t = u | X_{t-l}^{t-1} = \boldsymbol{w})\right)_{u \in \mathscr{X}}$.

Ainsi, cette représentation arborescente fournit une identification canonique entre les feuilles de l'arbre de dépendance et les paramètres de la chaîne de Markov.

L'idée des chaînes de Markov à longueur variable, telles que les a introduites J .RISSANEN ([**?**]), est de créer des modèles intermédiaires aux modèles de Markov classiques en élaguant les branches superflues de l'arbre de dépendence. Un exemple de tel arbre est présenté sur la figure 2.

3.2. Définition des arbres de contexte. Formellement, on définit une chaîne de Markov à longueur variable par une *fonction de contexte*, qui spécifie la longueur du contexte :

DÉFINITION 18. *Une application* $c : \mathscr{X}^l \to \cup_{i=0}^{l} \mathscr{X}^i$ *telle que :*

$$\forall \boldsymbol{w} = (w_l, \ldots, w_1) \in \mathscr{X}^l, \exists i \in \{1, \ldots, l\}, c(\boldsymbol{w}) = (w_i, \ldots, w_1)$$

est appelée une fonction de contexte*, et le mot* $c(\boldsymbol{w})$ *est appelé un* suffixe *de* $\boldsymbol{w}$.

Un arbre tel que celui représenté en figure 2 définit une fonction de contexte de la manière suivante : pour un mot $(w_l, \ldots, w_1)$, on parcourt l'arbre de la racine vers les feuilles, en suivant le chemin désigné par la séquence $(w_1, \ldots, w_l)$. Lorsque l'on rencontre une feuille, son chemin jusqu'à la racine est de la forme $\boldsymbol{w}' = (w_i, \ldots, w_1)$ pour i compris entre 1 et n. On pose alors $c(\boldsymbol{w}) = \boldsymbol{w}'$.

Il est alors immédiat de remarquer que l'image de la fonction c est identifiable à l'ensemble des feuilles de l'arbre de contexte, noté $\mathscr{C}(c)$. Cela implique en particulier que l'identification décrite ci-dessus entre arbres et fonction de contexte est biunivoque. On notera dans la suite indifféremment $\mathscr{C}(c)$ ou $\mathscr{C}(\tau)$ selon que le modèle est défini par un arbre ou sa fonction de contexte. Les éléments de cet ensemble sont appelés les *contextes de la chaîne de Markov à longueur variable.*

Puisque l'image d'un mot par la fonction de contexte est toujours un suffixe de ce mot, il suffit en fait de préciser sa longueur pour le définir de manière unique. Cela conduit à définir la *fonction de longueur* comme suit.

DÉFINITION 19. *On appelle* fonction de longueur *associée à la fonction de contexte c la fonction $l : \mathscr{X}^l \to \{1,\dots,l\}$ définie par la relation :*

$$\forall \boldsymbol{w} \in \mathscr{X}^l, l(\boldsymbol{w}) = |c(\boldsymbol{w})|$$

où $|c(\boldsymbol{w})|$ désigne la longueur du mot $c(\boldsymbol{w})$.

La fonction de longueur généralise donc la notion d'ordre d'une chaîne de Markov, ce dernier étant à présent dépendant des l lettres précédentes. Pour le voir, explicitons la définition d'une chaîne de Markov à longueur variable associée à la fonction de contexte c.

DÉFINITION 20. *Une chaîne de Markov d'ordre l $(X_t)_{t\in\mathbb{N}^\star}$ est une chaîne de Markov à longueur variable de fonction de contexte c si :*

$$\forall \boldsymbol{w} \in \mathscr{X}^l, \forall u \in \mathscr{X}, \forall t > l, \mathbb{P}(X_t = u | (X_{t-l},\dots,X_{t-1}) = \boldsymbol{w}) \\ = \mathbb{P}(X_t = u | (X_{t-l(\boldsymbol{w})},\dots,X_{t-1}) = c(\boldsymbol{w}))$$

Il apparaît alors que le vecteur de paramètres $\boldsymbol{\theta}$ d'une chaîne de Markov à longueur variable de fonction de contexte c est indexé par l'ensemble des contextes $\mathscr{C}(c)$, soit $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\boldsymbol{w}})_{\boldsymbol{w}\in\mathscr{C}(c)}$. Pour ce vecteur de paramètres, la vraisemblance du modèle s'écrit :

$$L(\boldsymbol{\theta}, \boldsymbol{x}) = \mu(x_1,\dots,x_l) \prod_{\boldsymbol{w}\in\mathscr{C}(c), u\in\mathscr{X}} \theta_{\boldsymbol{w}}^{N(\boldsymbol{w}u)}$$

où μ désigne la loi initiale de la séquence, et $N(\boldsymbol{w}u)$ le nombre d'occurences du mot $\boldsymbol{w}u$ dans la séquence $\boldsymbol{x}$.

3.3. Algorithme *context*. Nous abordons à présent la question de la sélection des dépendances *pertinentes* pour décrire les régularités d'une séquence observée, autrement dit de la sélection des observables. En introduisant les chaînes de Markov à longueur variable, J. RISSANEN définit une classe de modèles très vaste. Pour des raisons combinatoires évidentes, il est déraisonnable d'envisager réaliser une sélection de modèle gloutonne, par exemple en calculant le BIC de chacun des modèles.

Une telle classe de modèles est donc inutilisable en l'absence d'un algorithme de sélection de modèle efficace. Dans le cas des chaînes de Markov à longueur variable, cet algorithme a été introduit par RISSANEN, en même temps que la classe de modèles, et est connu sous le nom *context*. C'est un algorithme récursif, qui consiste à élaguer successivement les feuilles de l'arbre de contexte, tant que celles-ci ne modifient pas significativement la distribution sur la lettre qui suit.

Algorithme *context*

(1) Construire l'arbre de contexte de la chaîne de Markov d'ordre l,

(2) Itérer les étapes suivantes :

   (a) pour chaque feuille $\boldsymbol{w} = (w_i,\dots,w_1)$ de l'arbre, calculer la quantité :

$$S_{\boldsymbol{w}} = D\left(\hat{\mathbb{P}}(.|(X_{t-|\boldsymbol{w}|},\dots,X_{t-1}) = \boldsymbol{w}); \hat{\mathbb{P}}(.|(X_{t-|\boldsymbol{w}|+1},\dots,X_{t-1}) = \boldsymbol{w}')\right) \times N(\boldsymbol{w})$$

   (b) si $S_{\boldsymbol{w}} < K\log n$, où K est une constante et n la longueur de la séquence, on élague la feuille associée à $\boldsymbol{w}$.

   jusqu'à stabilité de l'arbre.

Dans les expressions précédentes, les quantités $\hat{\mathbb{P}}(.|X_{t-l},\dots,X_{t-1} = \boldsymbol{w})$ sont définies par :

$$\forall \boldsymbol{w} \in \mathscr{X}^l, \forall u \in \mathscr{X}, \hat{\mathbb{P}}(X_t = u | (X_{t-l},\dots,X_{t-1}) = \boldsymbol{w}) = \frac{N(\boldsymbol{w}u)}{N(\boldsymbol{w})}$$

3.4. Consistance. La principale contribution de P. BÜHLMANN et A. WYNER à la théorie des chaînes de Markov à longueur variable réside dans la preuve de consistance de l'algorithme *context* qu'il fournit dans [**?**]. Sous des hypothèses techniques que nous ne détaillons pas ici, ils montrent le résultat suivant.

THÉORÈME 27. *Soit $(X_t)_{1\leq t\leq n}$ une chaîne de Markov à longueur variable, et τ le plus petit arbre la représentant. Alors :*

$$\mathbb{P}(\hat{\tau}_n = \tau) \longrightarrow_{n\to\infty} 1$$

En termes de sous-modèles des chaînes de Markov, ce résultat montre que l'algorithme *context* sélectionne asymptotiquement le modèle le plus petit contenant la loi de la séquence.

La preuve de ce résultat distingue deux types de situation qui conduisent à l'estimation d'un arbre erroné :

– l'élagage d'une feuille qui est une feuille de l'arbre τ ayant donnée lieu aux données,
– l'absence d'élagage d'une feuille qui n'est pas présente dans l'arbre τ.

Dans le premier cas, on a d'une part la relation suivante qui traduit le fait que l'on a élagué la feuille uw :

$$D(\hat{\mathbb{P}}(.|uw)||\hat{\mathbb{P}}(.|w))N(uw) < K\log n \tag{33}$$

Par ailleurs, grâce à la condition de minimalité de l'arbre, et compte-tenu que uw est le label d'une feuille de τ, on a :

$$D(\mathbb{P}(.|uw)||\mathbb{P}(.|w)) > \varepsilon \tag{34}$$

pour $\varepsilon > 0$. Moyennant une hypothèse d'ergodicité du modèle, aisément assurée par l'hypothèse que toutes les transitions sont strictement positives, P. BÜHLMANN montre que pour tout $\alpha > 0$, alors pour n suffisamment grand :

$$\mathbb{P}((33)\,et\,(34)) < \alpha$$

Dans le second cas, la situation est exactement inversée. On a d'une part :

$$D(\hat{\mathbb{P}}(.|uw)||\hat{\mathbb{P}}(.|w))N(uw) > K\log n$$

et d'autre part :

$$\mathbb{P}(.|uw) = \mathbb{P}(.|w)$$

De la même manière, il construit une majoration de la probabilité de l'intersection ces deux événements similaire à la précédente. Le nombre de feuilles à envisager étant fini, ces deux majorations suffisent à majorer arbitrairement la probabilité d'estimer un arbre erroné, dès lors que la séquence est suffisamment longue.

3.5. Question de cohérence. La définition des modèles au moyen de la fonction de contexte soulève cependant un problème de cohérence statistique que l'on peut résumer ainsi : si l'on ne conserve qu'une feuille vw sous un nœud w, le modèle posé impose les relations :

$$\forall v' \in \mathscr{X}\backslash\{v\}, \forall u \in \mathscr{X}, \mathbb{P}(X_t = u|(X_{t-|w|-1},\ldots,X_{t-1}) = v'w) = \mathbb{P}(X_t = u|(X_{t-|w|},\ldots,X_{t-1}) = w) \tag{35}$$

$$\forall u \in \mathscr{X}, \mathbb{P}(X_t = u|(X_{t-|w|-1},\ldots,X_{t-1}) = vw) \neq \mathbb{P}(X_t = u|(X_{t-|w|},\ldots,X_{t-1}) = w) \tag{36}$$

Or, on a par ailleurs :

$$\forall u \in \mathscr{X}, \mathbb{P}(X_t = u|(X_{t-|w|-1},\ldots,X_{t-1}) = w) = \sum_{v'\in\mathscr{X}} \mathbb{P}(X_{t-|w|} = v')\mathbb{P}(X_t = u|(X_{t-|w|-1},\ldots,X_{t-1}) = v'w)$$

ce qui montre que :

$$\forall u \in \mathscr{X}, \mathbb{P}(X_t = u | (X_{t-|\boldsymbol{w}|-1}, \ldots, X_{t-1}) = v\boldsymbol{w}) = \mathbb{P}(X_t = u | (X_{t-|\boldsymbol{w}|}, \ldots, X_{t-1}) = \boldsymbol{w})$$

ce qui contredit la relation précédente. Ainsi définies, les chaînes de Markov à longueurs variables peuvent donc conduire à définir des modèles absurdes. Pour contourner cet écueil, P. BUHLMANN propose dans [**?**] une variante de la définition des VLMC, qui consiste à regrouper dans un nœud *virtuel* les nœuds que l'on élague. Nous reviendrons sur ce point dans la partie suivante, qui propose une généralisation de ces approches incluant, en particulier, des aspects de cette variante.

4. Arbres de contextes et compression

Indépendamment des résultats établis par BÜHLMANN quant à la consistance de l'algorithme *context*, les arbres de contexte ont principalement été développés en vue d'applications à la compression de texte. Deux algorithmes ont été dérivés des travaux initiaux de RISSANEN, et sont aujourd'hui extrêmement populaires : *context tree weighting* (pondération d'arbres de contexte), et *context tree maximization* (maximisation d'arbres de contexte). Introduits par WILLEMS, SHTARKOV et TJALKENS, ces méthodes combinent les qualités du codage adaptatif et des arbres de contexte. Nous les présentons ici, car ils constituent le point de départ des extensions originales proposées dans les parties suivantes.

4.1. Modèle bayésien. Combiner codage adaptatif et arbres de contexte revient, en pratique, à appliquer le schéma du codage adaptatif présenté précédemment à chacune des distributions conditionnelles du modèle de Markov à longueur variable associé. Soit τ un arbre de contexte de profondeur maximale l, défini par l'ensemble des contextes qui lui sont associés : $\tau \subset \cup_{k=1}^{l} \mathscr{A}^k$.

L'ensemble des raisonnements présentés dans le cadre des modèles d'indépendance pour construire le codage adaptatif peuvent être reproduits pour chacune des distributions $\boldsymbol{\theta}_{\boldsymbol{w}}$ associés aux contextes $\boldsymbol{w} \in \tau$. Si $\boldsymbol{x} \in \mathscr{A}^n$ désigne une séquence de longueur n, et $\boldsymbol{w}$ l'un des contextes de τ, on note $q_{\boldsymbol{w},t}$ la distribution de $\boldsymbol{\theta}_{\boldsymbol{w}}$ sachant la sous-séquence extraite $\boldsymbol{x}_{1,t}$ (on pose à nouveau que $q_{\boldsymbol{w},0}$ est une distribution de Dirichlet de paramètres $\boldsymbol{\alpha}$ identiques pour tous les contextes). De nouveau, on a la relation :

$$\forall \boldsymbol{\theta}_{\boldsymbol{w}} \in S_{|\mathscr{A}|}, q_{\boldsymbol{w},t+1}(\boldsymbol{\theta}_{\boldsymbol{w}}) = q_{\boldsymbol{w},t+1}(\boldsymbol{\theta}_{\boldsymbol{w}} | x_{t+1}) \mathbb{1}_{\{\boldsymbol{x}_{t-|\boldsymbol{w}|,t-1} = \boldsymbol{w}\}}$$

Cette relation traduit simplement le fait que la distribution des probabilités de transitions après le contexte $\boldsymbol{w} \in \tau$ n'est modifiée que lorsqu'une nouvelle observation est effectuée dans la séquence à la suite d'une occurrence de $\boldsymbol{w}$.

Par conséquent, pour un arbre τ fixé, la probabilité d'une séquence $\boldsymbol{x} \in \mathscr{A}^n$ sous le modèle bayésien associé s'écrit :

$$\mathbb{P}_\tau(\boldsymbol{x}) = \prod_{\boldsymbol{w} \in \tau} \frac{1}{B(\boldsymbol{\alpha})} \int_{\theta_{\boldsymbol{w}} \in S_{|\mathscr{A}|}} \prod_{a \in \mathscr{A}} \theta_{\boldsymbol{w},a}^{\alpha_a + N_{\boldsymbol{w}\,a}(\boldsymbol{x}) - 1}$$

4.2. Maximisation de $\mathbb{P}(\boldsymbol{x})$.

4.2.1. *Justification.* Intéressons-nous à présent au choix d'un *bon* arbre de contexte.

Si l'on s'intéresse à la compression d'un message, on va naturellement chercher à optimiser sa probabilité sous la loi de codage : on minimisera ainsi la longueur du message compressé. D'où le critère consistant à choisir le modèle sous lequel la probabilité du message est maximale, connu sous le nom de critère *MDL* (pour *Minimum Description Length*, [**?**]). Cette approche a été étudiée pour toutes les classes de modèle de type Markov : I. CSISZAR a ainsi introduit le critère BIC pour la sélection de modèle d'une chaîne de Markov, et démontré la consistance de la sélection obtenue (voir [**?, ?**]). Mais elle a été également mise en œuvre pour réaliser la sélection de modèles dans des

classes plus riches, telles que les arbres de contexte (voir [**?**]), et les résultats de consistance ont pu être dérivés.

Cependant, tout statisticien classique est interpellé par ce choix : lorsque l'on choisit le modèle qui explique le mieux la séquence par maximisation de la vraisemblance, on sait que l'on s'expose à la sur-estimation car la vraisemblance ne peut que croître lorsque l'on ajoute des paramètres à la distribution. La raison pour laquelle l'approche envisagée ici n'est pas exposée à ce risque tient au traitement bayésien du paramètre de la distribution de MARKOV. En effet, la sur-estimation survient lorsque l'augmentation du nombre de paramètres amène à diviser tellement l'échantillon entre les distributions à estimer (il y en a autant que de contextes, dans notre cas), que celles-ci sont estimées avec une confiance très faible. La prédiction des occurrences suivantes du même mot s'en ressentira, diminuant la probabilité finale de la séquence.

Ainsi, cette approche de la sélection de modèles *embarque* sa pénalisation par la prise en compte de l'incertitude sur le paramètre. Nous établirons dans la partie suivante une inégalité exponentielle pour la quantité $\mathbb{P}_\tau(\boldsymbol{x})$ lorsque τ n'est pas le plus petit modèle pour l'inclusion contenant la distribution supposée du message. Ceci nous permettra d'établir, dans un cadre un peu plus général, la consistance de cette procédure de sélection de modèle.

Intuitivement, le ressort de cette approche peut être formulé ainsi : lorsque le nombre d'observations après un contexte donné est faible, la distribution a posteriori du paramètre de la distribution après ce contexte n'est pas très concentrée autour de la distribution moyenne, que l'on obtiendrait par l'approche classique. Cela conduit à une probabilité moyenne (la moyenne référant à la distribution a posteriori du paramètre) de l'échantillon des lettres observées après ce contexte à laquelle contribuent des distributions conférant une probabilité plus faible à l'échantillon que la distribution de maximum de vraisemblance (qui représente la valeur maximale atteignable). Cette probabilité moyenne est donc d'autant plus faible. A l'inverse, avec un échantillon de lettres suivant le contexte très grand, la distribution a posteriori sur le paramètre sera très concentrée autour de la distribution de maximum de vraisemblance, et n'endommagera donc pas la probabilité moyenne de l'échantillon, qui asymptotiquement sera identique à celle obtenue par maximum de vraisemblance. Ainsi, cette approche réalise un compromis entre l'incertitude sur la valeur du paramètre due à la taille de l'échantillon utilisé pour l'estimer, et la capacité du modèle choisi à "bien" représenter les dépendances au sein de la séquence.

Formellement, ce compromis met en jeu l'expansion de LAPLACE, que nous verrons à l'œuvre dans la partie suivante. Intuitivement, cette opération permet de caractériser le comportement de la quantité $\int_{\theta\in\Theta_\tau}\mathbb{P}_\theta(\boldsymbol{x})\mathrm{d}q(\theta)$ en fonction de la valeur maximale atteinte par $\mathbb{P}_\theta(\boldsymbol{x})$ sur l'ensemble de distributions défini par le modèle τ, Θ_τ, et de la dimension de cet ensemble. Elle attribue ainsi un coût à la dimension du modèle considéré, qui correspond à la perte sur la probabilité moyenne $\int_{\theta\in\Theta_\tau}\mathbb{P}_\theta(\boldsymbol{x})\mathrm{d}q(\theta)$ due à la contribution des distributions de Θ_τ *éloignées* de la distribution de maximum de vraisemblance dans le même modèle. L'expansion de LAPLACE permet également de montrer que le critère BIC est une approximation du critère MDL.

4.2.2. *Algorithme CTM.* Ce choix de l'arbre maximisant la probabilité du message permet non seulement un apprentissage des dépendances à partir des symboles déjà transmis, mais cette opération peut, de plus, être effectuée *en ligne* avec une complexité algorithmique linéaire avec la longueur de la séquence.

Cette possibilité est justifiée par deux faits : d'abord, l'observation d'une nouvelle lettre demande simplement de mettre à jour les comptages d'un nombre fini de mots (ceux terminant par la lettre qui vient d'être observée, et d'une longueur inférieure ou égale à la profondeur de l'arbre plus 1). Ensuite, comme nous le détaillerons dans la

partie suivante dans un cadre un peu plus général, il est possible d'évaluer $\mathbb{P}_\tau(\boldsymbol{x})$ en recourant à une programmation dynamique ascendante dans l'arbre. Intuitivement, cela tient au fait que le choix optimal du sous-arbre sous un noeud donné ne dépend pas des choix effectués pour les sous-arbres qui n'ont aucun nœud en commun. Ainsi, un seul parcours de l'arbre de contexte plein (c'est-à-dire où tous les contextes apparaissant dans la séquence sont présents) est nécessaire.

Une présentation détaillée de CTM peut être trouvée dans [**?**].

Mentionnons également que le modèle bayésien peut être élargi, en probabilisant le choix du modèle. La probabilité d'une séquence devient alors :

$$\forall \boldsymbol{x} \in \mathscr{A}^n, \mathbb{P}(\boldsymbol{x}) = \sum_{\tau \in \mathscr{T}} \pi(\tau) \mathbb{P}_\tau(\boldsymbol{x})$$

Cette probabilité résulte donc d'une moyenne sur un ensemble de modèles $\mathscr{T}$, pondérée par une distribution a priori $(\pi(\tau))_{\tau \in \mathscr{T}}$, des probabilités de la séquence observée sous chacun des modèles envisagés. De ce point de vue, le choix du modèle τ conférant à la séquence la probabilité $\mathbb{P}_\tau(\boldsymbol{x})$ la plus grande revient à maximiser la probabilité a posteriori du modèle dans ce cadre bayésien étendu en utilisant un a priori uniforme sur l'ensemble des modèles. Cette équivalence n'est bien entendu valable que dans le cas d'un ensemble fini de modèles. Cependant, la probabilisation de l'ensemble des modèles rend possible de ne pas procéder à cette maximisation, et de considérer plutôt la probabilité de la séquence moyennée contre les modèles comme une probabilité de codage.

Cette approche est très populaire en compression de texte, et connue sous le nom de *Context Tree Weighting*. La construction de la méthode, ainsi que ses propriétés élémentaires vis-à-vis de la compression de texte, sont très bien décrits dans [**?**]. Cette approche, plus simple à mettre en œuvre que la maximisation de la probabilité de la séquence par le choix d'un arbre particulier, et quelque part plus générale (il n'y aucun choix arbitraire), est très populaire en compression de texte. Elle a donné lieu à de nombreux développements en aval des travaux de SHTARKOV, WILLEMS et TJALKENS, en particulier dans la direction d'une généralisation à des arbres de profondeur éventuellement infinie (voir [**?**] en particulier).

Troisième partie

Modèles de Markov parcimonieux, théorie

Introduction

Si les arbres de contexte constituent depuis plusieurs années un outil populaire dans le domaine de la compression, il n'en est pas de même dans le domaine de la bioinformatique. Seules quelques méthodes ont été publiées, dans le cadre de la prédiction de sites de fixation de facteurs de régulation ([**?**]*) d'une part, et aux fins de classification de protéines (*[**?**],[**?**]*). Remarquons aussi que des approches beaucoup plus générales ont été proposées pour le premier de ces problèmes, par exemple en substituant aux comptages des mots ceux de motifs impliquant des positions de la séquence possiblement éloignés (*[**?**]*). Il est cependant notable qu'aucune adaptation des méthodes afférentes aux chaînes de Markov cachées n'a été proposée pour permettre à celle-ci d'avoir des régimes* à longueur variable. *Disposer de telles méthodes permettrait en effet d'exploiter pleinement les potentialités des arbres de contexte, en particulier pour l'analyse bioinformatique des séquences de protéines.*

Cependant, l'application de ces modèles à la classification de protéines, même menée avec des outils puissants tels que les méthodes à noyau ([**?**]*), se heurte à la faible longueur des protéines. L'approche bayésienne, comme nous l'avons vu, pénalise l'ajout de paramètres en fonction de leur coût en termes de qualité d'estimation, et en l'occurrence les modèles permettant les meilleures prédictions sont, selon les auteurs, d'ordre faible. La taille de l'alphabet des acides aminés, comme la longueur typique des protéines (de quelques centaines à quelques milliers d'acides aminés) permettent en effet difficilement d'exploiter efficacement des modèles d'ordre supérieur à une ou deux unités.*

Dans le cas des protéines, les acides aminés peuvent présenter diverses similarités chimiques entre eux : certains peuvent se substituer à d'autres sans nuire à la fonctionnalité de la protéine. Sans être identiques pour autant, ces acides aminés se retrouvent également fréquemment substitués dans des protéines homologues issues de deux espèces pour des raisons évolutives. Pouvoir regrouper ces acides aminés, qui peuvent induire les mêmes contraintes sur leur voisinage dans la séquence, peut être une manière de capturer des dépendances d'ordre supérieur à 1 : en regroupant les contextes, l'information est moins partagée entre les distributions à estimer.

Formellement, prendre en compte ce type de regroupements revient à permettre la fusion de nœuds issus du même ancêtre dans les arbres de contexte. Le chapitre qui suit introduit le formalisme des modèles de Markov parcimonieux *comme une extension des arbres de contexte, dans laquelle ce type de fusion de contextes est permis. Nous munirons cette classe de modèles d'un modèle bayésien similaire à celui employé dans le cadre des arbres de contexte, et établirons la convergence de la sélection de modèle par maximum a posteriori vers le modèle le plus simple contenant la distribution génératrice des données. Dans la partie suivante, nous dériverons les équivalents des deux algorithmes classiques mettant en œuvre les arbres de contexte, CTM et CTW. Enfin, une évaluation de la qualité d'ajustement des modèles parcimonieux estimés à des séquences codantes sera présentée.*

CHAPITRE 6

Modèles de Markov parcimonieux

1. Définition

1.1. Fusionner plutôt qu'élaguer. Les arbres de contexte, ou chaînes de MARKOV à longueur variable, se sont révélés fournir de puissants outils de modélisation des séquences, et ce en particulier grâce au cadre bayésien (ou MDL dans le langage de la compression) qui permet de réaliser un compromis optimal entre la complexité du modèle (mesurée par le nombre de contextes différents pris en compte dans le modèle, autrement dit le nombre de feuilles de l'arbre de contextes) et la quantité d'informations apportée par la séquence pour estimer les transitions après chacun de ces contextes.

Cependant, tout optimal que soit ce compromis, il est relatif à l'ensemble de modèles envisagés. Or, si les arbres de contexte fournissent d'ores et déjà un enrichissement très conséquent de cet ensemble des modèles en comparaison aux chaînes de MARKOV d'ordre fixe, il est possible de proposer un ensemble radicalement plus grand de modèles. Il suffit pour cela de permettre que des nœuds issus d'un même parent dans l'arbre puissent être fusionnés. En permettant de telles fusions, on peut par exemple tirer profit de l'absence d'information apportée sur le symbole en position t, par exemple, par le fait de distinguer les nucléotides A et C lorsqu'elles apparaissent en position $t-1$, comme présenté en figure 1. Prendre en compte ce fait, et donc regrouper les échantillons correspondant à ces deux cas par la fusion des nœuds (et donc des sous-arbres sous ces nœuds) associés à A et T, permet en effet de disposer d'un échantillon plus grand pour estimer les probabilités de transition correspondantes, et ainsi d'améliorer la prédiction de la séquence.

Une première étape de ce projet consiste à formaliser l'ensemble de modèles obtenu en permettant ces fusions.

Ce faisant, nous introduisons des sous-modèles des chaînes de Markov vérifiant des relations d'égalité entre lignes de la matrice de transition plus complexes que dans le cas des chaînes de Markov à longueur variable. On attend par conséquent de ces modèles qu'ils fournissent des représentations plus parcimonieuses de la dépendance d'une chaîne de Markov.

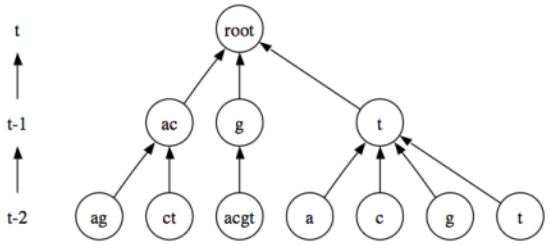


FIG. 1. Exemple d'arbre de contexte pour un modèle de Markov parcimonieux. L'alphabet est celui des nucléotides $\{a, c, g, t\}$.

1.1.1. *Motifs.* Afin de construire un formalisme permettant de travailler avec cette classe de modèles, nous introduisons une série de définitions. Tout d'abord, nous devons définir l'ensemble des labels des nœuds de l'*arbre de contexte parcimonieux*.

DÉFINITION 21. *On appelle* alphabet étendu *l'ensemble des parties non-vides de l'alphabet $\mathscr{A}$, et on le note $\bar{\mathscr{A}}$. Les éléments de l'alphabet étendu sont appelés* symboles, *et seront le plus souvent désignés par le caractère $\bar{u}$.*

Dès lors, on définit l'ensemble des chemins obtenus en concaténant les labels depuis une feuille jusqu'à la racine.

DÉFINITION 22. *Une séquence $\bar{\boldsymbol{w}} = (\bar{w}_1, \dots, \bar{w}_p) \in \bar{\mathscr{A}}^p$ est appelée un* p-motif, *ou simplement un* motif *s'il n'y a pas d'ambiguité sur sa longueur. L'ensemble des motifs de longueur p sera noté $\mathscr{C}_p$.*

Un motif étant une séquence de parties de l'alphabet, on peut définir une relation d'appartenance d'un mot à un motif de même longueur de la manière suivante :

DÉFINITION 23. *Un mot $\boldsymbol{w} = (w_1, \dots, w_p) \in \mathscr{A}^p$* appartient *au motif $\bar{\boldsymbol{w}} = (\bar{w}_1, \dots, \bar{w}_p)$ si et seulement :*

$$\forall i \in \{1, \dots, p\}, w_i \in \bar{w}_i$$

On note alors $\boldsymbol{w} \in \bar{\boldsymbol{w}}$.

1.1.2. *Fonctions de contextes parcimonieuses.* Nous avons alors tous les éléments pour définir les *fonctions de contexte parcimonieuses*, à partir desquelles nous allons définir les modèles. Les fonctions de contexte parcimonieuses généralisent les fonctions de contexte des chaînes de Markov à longueur variable. L'image d'un mot $\boldsymbol{w} \in \mathscr{A}^l$ n'en est plus un suffixe, mais un motif de même longueur : précisément le motif de $c(\mathscr{A}^l)$ tel que $w \in \bar{w}$. Formellement, cela conduit à la définition suivante des fonctions de contexte parcimonieuses.

DÉFINITION 24. *Une fonction $c : \mathscr{A}^l \to \mathscr{C}_l$ est une* fonction de contexte parcimonieuse *si :*

$$\forall \bar{\boldsymbol{w}} \in c(\mathscr{A}^l), c^{-1}(\bar{\boldsymbol{w}}) = \left\{ \boldsymbol{w} \in \mathscr{A}^l, \boldsymbol{w} \in \bar{\boldsymbol{w}} \right\}$$

Cette condition traduit le fait que la fonction de contexte *partitionne* l'ensemble des prédicteurs $\mathscr{A}^l$ de la chaîne de Markov d'ordre l sous la forme :

$$\mathscr{A}^l = \cup_{\bar{\boldsymbol{w}} \in c(\mathscr{A}^l)} \bar{\boldsymbol{w}}, \quad \text{avec } \forall \bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}' \in c(\mathscr{A}^l) \text{ tels que } \bar{\boldsymbol{w}} \neq \bar{\boldsymbol{w}}', \bar{\boldsymbol{w}} \cap \bar{\boldsymbol{w}}' = \emptyset$$

L'ensemble $c(\mathscr{A}^l)$ des *contextes* définis par une fonction de contexte parcimonieuse est également noté $\mathscr{C}(c)$.

1.1.3. *Fonctions et arbres de contextes parcimonieux.* Il nous reste alors à voir comment, dans le cas parcimonieux, on identifie un arbre avec l'ensemble des contextes. Nous allons désormais reprendre des notations dont les indices sont *renversés*, c'est-à-dire que les prédicteurs seront notés $\bar{\boldsymbol{w}} = (\bar{w}_l, \dots, \bar{w}_1)$, car ils sont des réalisations du l-uplet $(X_{t-l}, \dots, X_{t-l})$.

Soit donc une fonction de contexte parcimonieuse c, et considérons la racine de l'arbre. Afin de définir l'ensemble des symboles sous la racine, il faut utiliser les symboles utilisés en dernière position des contextes, et ainsi de suite jusqu'à chacune des feuilles. La proposition suivante garantit la cohérence de cette démarche.

PROPOSITION 3. *Pour tout motif $\bar{\boldsymbol{w}} = (\bar{w}_l, \dots, \bar{w}_1)$ de l'ensemble des contextes $\mathscr{C}(c)$, l'ensemble $\mathscr{M}_i(\bar{\boldsymbol{w}})$ défini par :*

$$\mathscr{M}_i(\bar{\boldsymbol{w}}) = \left\{ \bar{u}_{i+1}, \bar{\boldsymbol{u}} \in \mathscr{C}(c) \; et \; \forall 1 \leq j \leq i, \bar{u}_i = \bar{w}_i \right\}$$

est une partition de l'alphabet $\mathscr{A}$.

La preuve en est immédiate. Supposons qu'une lettre a_{i+1} n'appartienne à aucun des symboles de $\mathscr{M}_i(\bar{\boldsymbol{w}})$. Si $\boldsymbol{a}$ est un mot de i lettres tel que $\boldsymbol{a} \in (\bar{w}_i, \ldots, \bar{w}_1)$, alors tout mot terminant par $(a_{i+1}, \boldsymbol{a})$ n'a pas d'image par la fonction c, ce qui est absurde. De même, si une lettre a_{i+1} est présente dans deux symboles différents, les mots terminant par $(a_{i+1}, \boldsymbol{a})$ auront deux images par la fonction c. □

Ainsi, l'arbre associé à la fonction de contexte parcimonieuse c peut être construit récursivement, de la racine vers les feuilles, en ajoutant sous chaque nœud $\bar{\boldsymbol{w}}$ les symboles de $\mathscr{M}_i(\bar{\boldsymbol{w}})$.

Dans toute la suite du texte, nous désignerons un modèle de Markov parcimonieux indifféremment par son arbre τ ou sa fonction de contexte c.

1.2. Définition d'un modèle de Markov parcimonieux. Nous avons à présent tous les éléments pour définir les modèles de Markov parcimonieux. Suivant la démarche adoptée par P. BÜHLMANN, nous définissons une chaîne de Markov parcimonieuse par la factorisation de la vraisemblance que le modèle induit grâce à la fonction de contextes.

DÉFINITION 25. *Une chaîne de Markov d'ordre l $(X_t)_{t\in\mathbb{N}^\star}$ est une* chaîne de Markov parcimonieuse *de fonction de contexte c si :*

$$\forall \boldsymbol{w} \in \mathscr{A}^l, \forall u \in \mathscr{A}, \mathbb{P}(X_t = u | (X_{t-l}, \ldots, X_{t-1}) = \boldsymbol{w}) = \mathbb{P}\left(X_t = u | (X_{t-l}, \ldots, X_{t-1}) = c(\boldsymbol{w})\right)$$

Le vecteur de paramètres $\boldsymbol{\theta}_\tau$ d'une chaîne de Markov parcimonieuse est donc naturellement indexé par l'ensemble des motifs $\mathscr{C}(\tau)$ de l'arbre de contexte parcimonieux : $\boldsymbol{\theta}_\tau = (\boldsymbol{\theta}_{\bar{\boldsymbol{w}},u})_{\bar{\boldsymbol{w}}\in\mathscr{C}(\tau), u\in\mathscr{A}}$. Avec ces notations, la vraisemblance d'une séquence $\boldsymbol{x} = (x_1, \ldots, x_n)$ sous le modèle τ s'écrit :

$$L_\tau(\boldsymbol{\theta}_\tau, \boldsymbol{x}) = \mu(x_1, \ldots, x_l) \prod_{\bar{\boldsymbol{w}}\in\mathscr{C}(\tau)} \prod_{u\in\mathscr{A}} \boldsymbol{\theta}_{\bar{\boldsymbol{w}},u}^{N(\bar{\boldsymbol{w}}u)}$$

Afin d'illustrer le surcroît d'intérêt de ces modèles comparativement aux chaînes de Markov à longueur variable, nous proposons un exemple de matrice de transition d'ordre 1 qui ne donne lieu à aucun élagage dans le cadre des VLMC, mais qui donne lieu à deux fusions de deux paramètres. En effet, sur un alphabet à quatre lettres, la matrice de transition Π :

$$\Pi = \begin{pmatrix} 0,2 & 0,2 & 0,3 & 0,3 \\ 0,2 & 0,2 & 0,3 & 0,3 \\ 0,3 & 0,3 & 0,2 & 0,2 \\ 0,3 & 0,3 & 0,2 & 0,2 \end{pmatrix}$$

est encodée par un arbre plein dans le cadre des VLMC, mais par un arbre dans lequel les deux premières lettres de l'alphabet sont fusionnées, de même que les deux dernières. Dans cet exemple, le recours aux modèles parcimonieux permet donc l'économie de 6 paramètres libres.

On peut alors s'assurer que l'ensemble des modèles de Markov parcimonieux contient l'ensemble des chaînes de Markov à longueur variable. En effet, donnons nous un arbre VLMC τ, et notons l la profondeur de sa plus longue branche. On complète alors τ de manière à ce que l'ensemble de ses branches ait la longueur l, en ajoutant sous chaque noeud la partition $\mathscr{A}$, autrement dit la partition dans laquelle l'ensemble des lettres sont regroupées. On obtient ainsi un arbre qui représente un modèle de Markov parcimonieux, et qui induit les mêmes contraintes sur la matrice de transition que l'arbre VLMC τ original.

Qui plus est, on montre aisément que la plupart des modèles parcimonieux ne peuvent être représentés comme des VLMC. Il suffit en effet pour cela que :

- l'arbre parcimonieux utilise des partitions à plus de deux éléments sous l'un de ses noeuds,
- présente des fusions pour au moins un nœud interne, sans que ses noeuds enfants ne soient eux-mêmes fusionnés.

Les modèles de Markov parcimonieux constituent par conséquent une classe de modèles strictement plus grande que les VLMC. Ce point est la principale motivation à leur introduction : moyennant de disposer d'un critère de qualité d'ajustement d'un modèle à des données, on est assuré que le modèle parcimonieux optimal sous ce critère est meilleur que le modèle VLMC optimal, la comparaison ayant lieu sous le même critère.

Mais avant de nous atteler à la recherche d'un tel critère, quelques opérations de dénombrement permettent de juger de l'ampleur de la généralisation effectuée.

1.3. Combinatoire. Les généralisations introduites pour construire les modèles de Markov parcimonieux ont des conséquences en termes combinatoires : le nombre de modèles parcimonieux d'un ordre donné sur un alphabet croît très rapidement avec l'ordre, y compris pour des alphabets de petites tailles. La combinatoire des modèles parcimonieux résulte de la composition de la combinatoire des partitions de l'alphabet avec celle du nombre de nœuds d'un arbre.

1.3.1. *Nombre de partitions de l'alphabet.* Le nombre de partitions B_n d'un ensemble $\mathscr{A}$ de cardinal n est connu comme le *nombre de Bell*. Ces nombres ont été largement étudiés depuis un article de DOBINSKI [**?**], dans lequel la formule suivante est établie :

$$\mathscr{N}(\mathscr{A}) = \frac{1}{e}\sum_{i=0}^{\infty}\frac{i^{|\mathscr{A}|}}{i!} \tag{37}$$

DOBINSKI établit cette formule en décomposant l'ensemble des partitions en la réunion des partitions de même taille k. Pour un alphabet de taille n, le nombre de partitions de taille k s'appelle *nombre de Stirling de seconde espèce* $B_{n,k}$. Ces nombres vérifient la récurrence suivante :

$$B_{n,k} = B_{n-1,k-1} + kB_{n-1,k}$$

qui découle de la simple disjonction de cas selon que le $n^{\grave{e}}$ élément est partitionné en un singleton, ou qu'il appartient à une partie plus large. Sommer cette récurrence sur k, puis vérifier que la formule 37 est valide représente un calcul fastidieux. J. PITMAN dérive cette formule d'une manière extrêmement simple dans [**?**].

Le point essentiel est que la relation 37 montre une croissance exponentielle du nombre de partitions avec la taille de l'alphabet. Cette vitesse extrêmement rapide sera la source d'écueils combinatoires dans les algortihmes de sélection de modèles, dans la mesure où l'énumération répétée de l'ensemble des partitions possibles de l'alphabet sera alors requise.

Taille de l'alphabet	3	4	5	10	20
Nb. partitions	5	15	52	115975	5.17×10^{13}

FIG. 2. Le nombre de partitions de l'alphabet, pour des tailles d'alphabet de 3 à 20

1.3.2. *Nombre de modèles.* Le résultat précédent est crucial dans le dénombrement des modèles parcimonieux. Si le nombre de modèles d'un ordre donné est difficile à calculer analytiquement, on peut cependant dériver de la structure d'arbre un algorithme récursif qui permet ce calcul en un temps raisonnable, dont nous présenterons les résultats par la suite.

Nous présentons d'abord un raisonnement rapide permettant une majoration grossière du nombre de modèles : pour construire un arbre, il faut choisir une partition qui définit ses au plus $|\mathscr{A}|$ fils, puis celles sous chacun d'eux, et ainsi de suite. En majorant le nombre de fils sous un nœud par la taille de l'alphabet, il vient :

$$\mathscr{N}(\mathscr{A}, k) = B_{|\mathscr{A}|}\mathscr{N}(\mathscr{A}, k-1)^{\mathscr{A}}, \quad k = 2, \ldots, l$$

$\lvert\mathscr{A}\rvert$	Ordre 1	2	3	4	5	6
3	5	205	$8,7\times10^{6}$	$6,7\times10^{20}$	3×10^{62}	$2,6\times10^{187}$
4	15	72465	$2,8\times10^{19}$	$5,8\times10^{77}$	*overflow*	*overflow*
5	52	$4,6\times10^{8}$	2×10^{43}	$3,1\times10^{216}$	*overflow*	*overflow*
6	203	$7,5\times10^{13}$	$1,8\times10^{83}$	*overflow*	*overflow*	*overflow*
7	877	$4,1\times10^{20}$	$1,9\times10^{144}$	*overflow*	*overflow*	*overflow*
8	4140	$8,7\times10^{28}$	$3,2\times10^{231}$	*overflow*	*overflow*	*overflow*
9	21147	$8,5\times10^{38}$	*overflow*	*overflow*	*overflow*	*overflow*
10	115975	$4,4\times10^{50}$	*overflow*	*overflow*	*overflow*	*overflow*
11	678570	$1,4\times10^{64}$	*overflow*	*overflow*	*overflow*	*overflow*
12	$4,2\times10^{6}$	$3,1\times10^{79}$	*overflow*	*overflow*	*overflow*	*overflow*
13	$2,8\times10^{7}$	$5,5\times10^{96}$	*overflow*	*overflow*	*overflow*	*overflow*
14	$1,9\times10^{8}$	$8,5\times10^{115}$	*overflow*	*overflow*	*overflow*	*overflow*
15	$1,4\times10^{9}$	$1,3\times10^{137}$	*overflow*	*overflow*	*overflow*	*overflow*
16	1×10^{10}	$2,1\times10^{160}$	*overflow*	*overflow*	*overflow*	*overflow*
17	$8,3\times10^{10}$	$4,1\times10^{185}$	*overflow*	*overflow*	*overflow*	*overflow*
18	$6,8\times10^{11}$	1×10^{213}	*overflow*	*overflow*	*overflow*	*overflow*
19	$5,8\times10^{12}$	$3,6\times10^{242}$	*overflow*	*overflow*	*overflow*	*overflow*
20	$5,2\times10^{13}$	$1,9\times10^{274}$	*overflow*	*overflow*	*overflow*	*overflow*

FIG. 3. Le nombre de modèles pour des alphabets de taille 3 à 20, et des ordres de 1 à 5. La précision machine est dépassée pour les entrées *overflow*. Malgré le recours à une échelle logarithmique, on néglige un grand nombre de fois (compte-tenu du caractère récursif du calcul) des petites quantités, sans contrôle possible sur ces approximations.

ce qui induit un équivalent du nombre de modèles :

$$\mathscr{N}(\mathscr{A}, l) \sim \exp \lambda |\mathscr{A}|^{l}$$

où λ est une constante strictement positive.

Le tableau 3 résume les valeurs du nombre de modèles pour différentes tailles de l'alphabet et différents ordres.

On vérifie alors empiriquement que le nombre de modèles est de l'ordre de $\exp(|\mathscr{A}|^{l})$, où l désigne la profondeur des arbres envisagés. La figure 4 représente la racine carrée du lagarithme du nombre de modèles de profondeur 2, en fonction de la taille de l'alphabet.

L'introduction de la possibilité de fusionner les nœuds se traduit par conséquent par une explosion du nombre de modèles possibles, l'alphabet et l'ordre étant fixés. En effet, dans le cas des VLMC, le nombre de configurations possibles des nœuds sous un nœud interne est proportionnel à la taille de l'alphabet, et non pas exponentiel en cette quantité.

2. Modèle bayésien

2.1. Définitions. Nous emploierons par la suite les notations suivantes, pour un modèle τ et une valeur $\boldsymbol{\theta}_\tau$ du paramètre dans ce modèle :

- $\Pi(\tau)$ désigne la distribution de probabilités a priori sur les modèles,
- $\pi(\boldsymbol{\theta}_\tau|\tau)$ désigne la densité de probabilité a priori sur les paramètres dans le modèle τ.

L'ordre des modèles envisagés dans la suite est fixé à l. Pour la lisibilité des formules, ce paramètre ne figurera pas en indice.

2.1.1. *Loi d'une séquence.* Le modèle bayésien a priori formalise alors le modèle aléatoire consistant à :

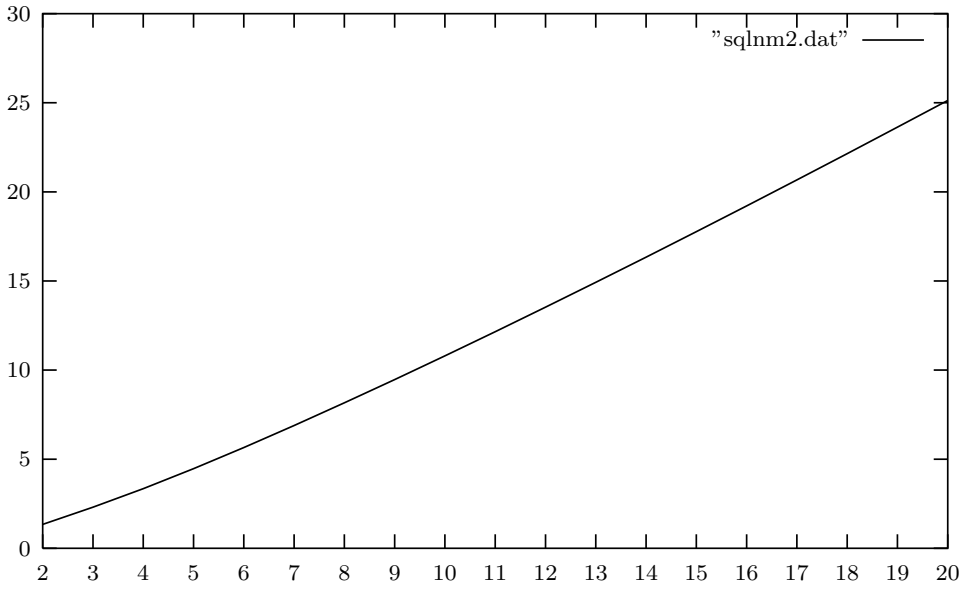


FIG. 4. Racine carrée du logarithme du nombre de modèles de profondeur 2, en fonction de la taille de l'alphabet.

– tirer un modèle τ selon la loi $\Pi(\tau)$,
– tirer un paramètre $\boldsymbol{\theta}_\tau$ selon la loi $\pi(\boldsymbol{\theta}_\tau|\tau)$,
– tirer une séquence $\boldsymbol{x}$ selon la distribution $\mathbb{P}_{\boldsymbol{\theta}_\tau,\tau}(\boldsymbol{x})$.

On définit ainsi la distribution a priori suivante sur les séquences $\boldsymbol{x} \in \mathscr{A}^n$:

$$\mathbb{P}(\boldsymbol{x}) = \sum_{\tau\in\mathscr{T}_l} \Pi(\tau) \int_{\Theta_\tau} \Big(\prod_{\bar{\boldsymbol{w}}\in\mathscr{C},u\in\mathscr{A}} \theta_\tau(\bar{\boldsymbol{w}},u)^{N(\bar{\boldsymbol{w}}u)} \Big) \pi(\boldsymbol{\theta}_\tau|\tau)\,\mathrm{d}\boldsymbol{\theta}_\tau \tag{38}$$

2.2. Modèle bayésien sur les paramètres.

2.2.1. *Conditionnement par le modèle.* En toute généralité, il faudrait prévoir une distribution différente des paramètres dans chacun des modèles. Cependant, compte-tenu du nombre de modèles, ce n'est pas une démarche tractable. Aussi, nous prenons en compte le fait que chaque modèle parcimonieux d'ordre l est un sous-modèle linéaire du modèle de Markov d'ordre l. Pour cela, nous ne définissons que la distribution $\pi(\boldsymbol{\theta})$ sur le paramètre du modèle de Markov d'ordre l. Elle induit alors une densité sur les paramètres d'un modèle τ de la manière suivante :

$$\pi(\boldsymbol{\theta}_\tau|\tau) = \frac{\pi(\boldsymbol{\theta})}{\int_{\Theta_\tau} \pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}} \tag{39}$$

où $\boldsymbol{\theta}_\tau$ est canoniquement plongé dans le modèle Θ par la relation :

$$\forall \boldsymbol{w} \in \mathscr{A}^l, \forall u \in \mathscr{A}, \theta_{\boldsymbol{w},u} = \theta_{\tau,c_\tau(\boldsymbol{w},}$$

Ainsi, la densité du paramètre dans le modèle τ est la trace de la densité du paramètre de la chaîne de Markov.

Au-delà de la simplification drastique du modèle qu'elle introduit, cette contrainte est cependant justifiée par le fait que la sélection de modèle s'attache finalement à mieux estimer la chaîne de Markov d'ordre l.

Cette approche a été proposée dans [**?**], dans le cadre des méthodes de pondération d'arbres de contextes pour la compression de texte. Elle est également reprise dans [**?**] pour la construction d'un noyau pour séquences. Mais nous reviendrons sur ces approches alternatives dans la dernière partie.

2.3. Distribution a priori sur les paramètres. Il reste alors à choisir la forme que nous souhaitons donner à la distribution a priori sur le paramètre de la chaîne de Markov. Le paramètre $\boldsymbol{\theta}$ d'une chaîne de Markov intervient dans la vraisemblance sous la forme d'un produit :

$$L(\boldsymbol{\theta}) = \prod_{\boldsymbol{w}\in\mathscr{A}^l, u\in\mathscr{A}} \theta_\tau(\boldsymbol{w},u)^{N(\boldsymbol{w}u)} \tag{40}$$

Afin d'obtenir des formules analytiques pour les distributions a posteriori, il convient de choisir un modèle a priori sur les paramètres tel que le produit d'une distribution de ce modèle avec la vraisemblance 40 est également une distribution du modèle. Cette propriété, dite de conjugaison entre la distribution sur le paramètre et la vraisemblance, permet d'obtenir des formules simples pour le calcul de la distribution a posteriori.

Dans le cas de la vraisemblance d'une chaîne de Markov, le modèle a priori sur les paramètres qui présente cette propriété est celui constitué des distributions de Dirichlet $\mathscr{D}\big((\alpha_{\boldsymbol{w},u})_{\boldsymbol{w}\in\mathscr{A}^l,u\in\mathscr{A}}\big)$, dont la densité $\pi(\boldsymbol{\theta})$ s'écrit :

$$\pi(\boldsymbol{\theta}) = \prod_{\boldsymbol{w}\in\mathscr{A}^l} \Gamma\Big(\sum_{u\in\mathscr{A}} \alpha_{\boldsymbol{w},u}\Big) \prod_{u\in\mathscr{A}} \frac{\theta(\boldsymbol{w},u)^{\alpha_{\boldsymbol{w},u}-1}}{\Gamma(\alpha_{\boldsymbol{w},u})} \tag{41}$$

Le produit de la vraisemblance avec la densité a priori de la chaîne de Markov s'écrit alors :

$$L(\boldsymbol{\theta},\boldsymbol{x})\pi(\boldsymbol{\theta}) = \prod_{\boldsymbol{w}\in\mathscr{A}^l} \Gamma(\sum_{u\in\mathscr{A}} \alpha_{\boldsymbol{w},u}) \prod_{u\in\mathscr{A}} \frac{\theta(\boldsymbol{w},u)^{N(\boldsymbol{w}u)+\alpha_{\boldsymbol{w},u}-1}}{\Gamma(\alpha_{\boldsymbol{w},u})}$$

Les distributions sur les paramètres des modèles parcimonieux préconisées au paragraphe précédent sont alors aisées à exprimer. En effet, lorsque $\boldsymbol{\theta} \sim \mathscr{D}(\boldsymbol{\alpha})$, avec $\boldsymbol{\alpha} = (\alpha_{\boldsymbol{w},u})_{(\boldsymbol{w},u)\in\mathscr{A}^{l+1}}$, alors quelque soit l'arbre τ, la contrainte (39) se traduit par :

$$p(\boldsymbol{\theta}_\tau|\tau) = C \prod_{(\bar{\boldsymbol{w}},u)\in\mathscr{C}(\tau)\times\mathscr{A}} \theta_{\tau,\bar{\boldsymbol{w}},u}^{(\sum_{\boldsymbol{w}\in\bar{\boldsymbol{w}}}\alpha_{\boldsymbol{w},u})-|\bar{w}|}$$

Il suffit donc de définir les quantités $\alpha_{\bar{\boldsymbol{w}},u}$ de la manière suivante :

$$\forall \bar{\boldsymbol{w}} \in \mathscr{C}_l, \forall u \in \mathscr{A}, \alpha_{\bar{\boldsymbol{w}},u} = (\sum_{\boldsymbol{w}\in\bar{\boldsymbol{w}}} \alpha_{\boldsymbol{w},u}) - |\bar{\boldsymbol{w}}| + 1 \tag{42}$$

pour que la vraisemblance moyenne dans un modèle τ s'écrive :

$$\int_{\Theta_\tau} L_\tau(\boldsymbol{\theta}_\tau|\boldsymbol{x})\pi(\boldsymbol{\theta}_\tau|\tau)\,\mathrm{d}\boldsymbol{\theta}_\tau = \prod_{\bar{\boldsymbol{w}}\in\mathscr{C}(\tau)} \frac{\Gamma(\sum_{u\in\mathscr{A}}\alpha_{\bar{\boldsymbol{w}},u})}{\Gamma(\sum_{u\in\mathscr{A}} N(\bar{\boldsymbol{w}}u)+\alpha_{\bar{\boldsymbol{w}},u})} \prod_{u\in\mathscr{A}} \frac{\Gamma(N(\bar{\boldsymbol{w}}u)+\alpha_{\bar{\boldsymbol{w}},u})}{\Gamma(\alpha_{\bar{\boldsymbol{w}},u})} \tag{43}$$

Il faut cependant respecter la contrainte suivante, afin que les distributions traces soient correctement définies :

$$\forall \bar{\boldsymbol{w}} \in \mathscr{C}_l, \forall u \in \mathscr{A}, \frac{\sum_{\boldsymbol{w}\in\bar{\boldsymbol{w}}}\alpha_{\boldsymbol{w},u}}{|\bar{\boldsymbol{w}}|} > 1 - \frac{1}{|\bar{\boldsymbol{w}}|} \tag{44}$$

Dans le cas où l'on choisit des paramètres $\alpha_{\boldsymbol{w},u}$ indépendants de $\boldsymbol{w}$ et u, il faudra donc choisir :

$$\forall w \in \mathscr{C}_l, \forall u \in \mathscr{A}, \alpha_{\boldsymbol{w},u} > 1 - \frac{1}{|\boldsymbol{X}|^l}$$

afin que toutes les densités trace soient correctement définies. Dans le cas particulier d'une chaîne de Markov d'ordre 1 sur un alphabet binaire, le cas limite s'écrit :

$$\forall \boldsymbol{w} \in \mathscr{A}^l, \forall u \in \mathscr{A}, \alpha_{\boldsymbol{w},u} = \frac{1}{2}$$

Cette distribution est connue comme le prior de KRICHEVSKY-TROFIMOV, largement utilisée en théorie de l'information pour définir des *lois de codage*, comme nous le verrons dans la partie consacrée aux approches informationnelles.

Notons enfin que l'on rencontre fréquemment un paramétrage translaté de la distribution de Dirichlet, sous la forme des $\alpha'_{\boldsymbol{w},u} = \alpha_{\boldsymbol{w},u} - 1$. On remarque alors que la

manière dont les paramètres $\boldsymbol{\alpha}'$ interviennent dans la vraisemblance conduit à une interprétation des quantités $N(\bar{\boldsymbol{w}}u)+\alpha'_{\bar{\boldsymbol{w}},u}$ comme des *pseudo-comptages*. En ces termes, la relation (42) se réécrit :

$$\alpha'_{\bar{\boldsymbol{w}},u}=\sum_{\boldsymbol{w}\in\bar{\boldsymbol{w}}}\alpha'_{\boldsymbol{w},u}$$

Elle est tout à fait naturelle, puisque les comptages répondent à la relation :

$$\forall\bar{\boldsymbol{w}}\in\mathscr{C}_l, N(\bar{\boldsymbol{w}}u)=\sum_{\boldsymbol{w}\in\bar{\boldsymbol{w}}}N(\boldsymbol{w}u)$$

En pratique, nous préférerons le choix d'une distribution uniforme sur le domaine du paramètre, par le choix des paramètres a priori :

$$\forall\boldsymbol{w}\in\mathscr{A}^l,\forall u\in\mathscr{A},\alpha_{\boldsymbol{w},u}=1$$

Ce choix est de plus conservatif dans le passage au sous-modèle, au sens où :

$$\forall\bar{\boldsymbol{w}}\in\mathscr{C}_l,\alpha_{\bar{\boldsymbol{w}},u}=1$$

si l'on applique la relation 39. Ceci est naturel puisqu'il représente le choix de pseudo-comptages nuls.

2.4. Distribution a posteriori. Il est à présent possible de dériver la densité $p(\boldsymbol{\theta}_\tau|\tau,\boldsymbol{x})$ de la distribution a posteriori du paramètre dans un modèle τ. D'après la formule de BAYES, elle s'écrit :

$$p(\boldsymbol{\theta}_\tau|\tau,\boldsymbol{x})=\frac{L(\boldsymbol{\theta}_\tau,\boldsymbol{x})\pi(\boldsymbol{\theta}_\tau|\tau)}{\mathbb{P}(\boldsymbol{x},\tau)}$$

d'où la relation :

$$p(\boldsymbol{\theta}_\tau|\tau,\boldsymbol{x})=\frac{\prod_{\bar{\boldsymbol{w}}\in\mathscr{C}(\tau)}\prod_{u\in\mathscr{A}}\theta_{\bar{\boldsymbol{w}},u}^{N(\bar{\boldsymbol{w}}u)+\alpha_{\bar{\boldsymbol{w}},u}-1}}{\int_{\Theta_\tau}\prod_{\bar{\boldsymbol{w}}\in\mathscr{C}(\tau)}\prod_{u\in\mathscr{A}}\theta_{\bar{\boldsymbol{w}}}^{N(\bar{\boldsymbol{w}}u)+\alpha_{\bar{\boldsymbol{w}}u}}\,\mathrm{d}\boldsymbol{\theta}_\tau}$$

Il en découle que la distribution a posteriori du paramètre est également une loi de Dirichlet, dont les paramètres sont définis de la manière suivante :

$$\forall\bar{\boldsymbol{w}}\in\mathscr{C}(\tau),\forall u\in\mathscr{A},\hat{\alpha}_{\bar{\boldsymbol{w}},u}=N(\bar{\boldsymbol{w}}u)+\alpha_{\bar{\boldsymbol{w}},u}$$

La relation similaire pour les paramètres translatés α' s'écrit de manière identique. Cette relation permet d'établir la propriété suivante :

PROPOSITION 4. *Si les distributions a priori sur les paramètres des différents modèles sont des distributions de* Dirichlet *construites conformément à (*39*), la distribution a posteriori du paramètre d'un modèle τ s'obtient également comme la trace de la densité a posteriori du paramètre du modèle de Markov d'ordre l.*

La preuve de cette proposition résulte immédiatement de la relation :

$$\forall\bar{\boldsymbol{w}}\in\mathscr{C}_l,\forall u\in\mathscr{A},N(\bar{\boldsymbol{w}}u)=\sum_{\boldsymbol{w}\in\bar{\boldsymbol{w}}}N(\boldsymbol{w}u)$$

et de la relation (39).

On déduit également de cette relation l'expression de l'estimateur par maximum a posteriori.

PROPOSITION 5. *L'estimateur par maximum a posteriori $\tilde{\boldsymbol{\theta}}_\tau$ des transitions du modèle de Markov parcimonieux s'écrit :*

$$\forall\tau\in\mathscr{T}_l,\forall\bar{\boldsymbol{w}}\in\mathscr{C}(\tau),\forall u\in\mathscr{A},\tilde{\theta}_{\bar{\boldsymbol{w}}u}=\frac{N(\bar{\boldsymbol{w}}u)+\alpha_{\bar{\boldsymbol{w}},u}-1}{\sum_{u\in\mathscr{A}}N(\bar{\boldsymbol{w}}u)+\alpha_{\bar{\boldsymbol{w}}u}-1}$$

En utilisant le paramétrage translaté $\alpha'_{\boldsymbol{w},u} = \alpha_{\boldsymbol{w},u} + 1$, le maximum a posteriori s'écrit :

$$\tilde{\theta}_{\tilde{\boldsymbol{w}}u} = \frac{N(\tilde{\boldsymbol{w}}u) + \alpha'_{\tilde{\boldsymbol{w}},u}}{\sum_{u\in\mathscr{A}} N(\tilde{\boldsymbol{w}}u) + \alpha'_{\tilde{\boldsymbol{w}}u}}$$

On remarque alors que le choix d'une distribution a priori uniforme sur le paramètre entraîne l'égalité du maximum a posteriori et du maximum de vraisemblance.

2.5. Distribution sur les modèles. Autant la probabilisation du paramètre d'une distribution au sein d'un modèle fixé peut trouver sa justification dans les fondements de la statistique bayésienne tels qu'ils ont été présentés dans la partie 2, autant la même opération sur l'ensemble des modèles est délicate. Tout d'abord, l'intuition introduite dans la partie 2, selon laquelle le recours à cette probabilisation correspond à une forme de consistance du raisonnement probabiliste, semble beaucoup plus difficile à appliquer. Nous n'avons, pour la sélection de modèles, pas d'équivalent au principe de maximum d'entropie, et donc pas de manière naturelle de choisir une distribution particulière pour décrire un macro-état d'un système. Et donc pas de point de départ pour mener un raisonnement similaire ici.

Cependant, il reste vrai que quelque soit la procédure de sélection de modèle utilisée, une autre réalisation de l'aléa aurait pu conduire à la sélection d'un modèle différent. C'est d'ailleurs cette incertitude sur le modèle sélectionné que permet de prendre en compte l'algorithme CTW. Dans la littérature des arbres de contexte, un choix fréquent est de considérer que l'arbre peut se terminer au nœud courant avec une probabilité 1/2, ou être encore plus profond avec la même probabilité. Mais ce choix ne reçoit aucune réelle justification fondamentale. Il présente simplement l'avantage (partagé avec de nombreuses autres distributions a priori envisageables) d'admettre une somme finie sur l'ensemble des modèles, y compris si ceux-ci sont en nombre infini (ce que nous n'envisagerons pas ici). Nous proposons ici divers choix de distribution a priori, puis dérivons les distributions a posteriori auxquels ils conduisent.

2.5.1. *Distribution a priori sur les modèles.* Probabiliser l'ensemble des modèles soulève Parmi les pénalités envisageables, la plus simple est de la forme suivante :

$$\Pi(\tau) = C(k)\frac{1}{k^{|\mathscr{C}(\tau)|}}$$

car elle s'écrit également sous la forme :

$$\Pi(\tau) = C(k)\prod_{\tilde{\boldsymbol{w}}\in\mathscr{C}(\tau)}\frac{1}{k}$$

On dispose ainsi d'une gamme de pénalités, paramétrée par la constante k, dont le choix déplace la distribution a priori du nombre de paramètres : inférieure à 1, elle favorise les arbres comprenant beaucoup de feuilles, supérieure à 1, elle les défavorise. Nous discuterons le choix de cette pénalité au travers de simulations dans la dernière partie. Cette gamme de pénalités répond aux contraintes suivantes :

- elle s'écrit comme un produit sur les feuilles (nous allons voir dès le paragraphe suivant comment traiter la constance de normalisation $C(k)$ pour que ce soit le cas),
- elle pénalise d'autant plus un arbre que le nombre de ses feuilles est élevé.

Normalisation de la distribution. Nous nous intéressons à présent au problème de la normalisation de la distribution sur les modèles. Nous devons avoir :

$$\sum_{\tau\in\mathscr{T}_l}\Pi(\tau) = 1$$

soit :

$$C(k) = \frac{1}{\sum_{\tau\in\mathscr{T}_l}\Pi(\tau)}$$

Nous verrons dans la partie suivante un algorithme de programmation dynamique permettant l'évaluation de cette constante, aussi bien dans le cadre de la distribution a priori que dans le cadre de la distribution a posteriori. Notre propos ici est plutôt de voir comment cette constante $C(k)$ peut être également mise sous la forme d'un produit :

$$C(k) = \prod_{\bar{w} \in \mathscr{C}(\tau)} c_{\bar{w}(\tau)}$$

L'idée pour ce faire est que pour tout arbre τ de profondeur l, l'ensemble $\mathscr{C}(\tau)$ des contextes définit une partition de l'ensemble des mots de longueur l :

$$\mathscr{A}^l = \cup_{\bar{w} \in \mathscr{C}(\tau)} \bar{w}$$

où un motif $\bar{w}$ est vu comme l'ensemble des mots qu'il représente. De cette décomposition de l'alphabet en parties disjointes, on déduit :

$$|\mathscr{A}|^l = \sum_{\bar{w} \in \mathscr{C}(\tau)} |\bar{w}|$$

où $|\bar{w}|$ est le nombre de mots qui représentés par le motif $\bar{w}$.

Par conséquent, en posant :

$$c_{\bar{w}}(k) = C(k)^{\frac{|\bar{w}|}{|\mathscr{A}|^l}}$$

on a :

$$\forall \tau \in \mathscr{T}_l, \prod_{\bar{w} \in \mathscr{C}(\tau)} c_{\bar{w}}(k) = C(k)$$

Finalement, la probabilité d'une séquence sous le modèle a priori s'écrit, en fonction des paramètres $(\alpha_{\bar{w},u})_{\bar{w} \in \mathscr{C}_l, u \in \mathscr{A}}$ et k :

$$\mathbb{P}(\boldsymbol{x}) = \sum_{\tau \in \mathscr{T}_l} \prod_{\bar{w} \in \mathscr{C}(\tau)} \frac{c_{\bar{w}}(k) \times \Gamma(\sum_{u \in \mathscr{A}} \alpha_{\bar{w},u})}{k \times \Gamma(\sum_{u \in \mathscr{A}} \alpha_{\bar{w},u} + N(\bar{w}u))} \prod_{u \in \mathscr{A}} \frac{\Gamma(\alpha_{\bar{w},u} N(\bar{w}u))}{\Gamma(\alpha_{\bar{w},u})} \tag{45}$$

2.5.2. *Distribution a posteriori sur les modèles.* Il est alors aisé de dériver la distribution a posteriori sur les modèles, sachant une séquence observée $\boldsymbol{x} = (x_1, \ldots, x_n)$. En effet, d'après la formule de Bayes :

$$\mathbb{P}(\tau | \boldsymbol{x}) = \frac{\mathbb{P}(\tau, \boldsymbol{x})}{\mathbb{P}(\boldsymbol{x})}$$

d'où :

$$\mathbb{P}(\tau | \boldsymbol{x}) = \frac{1}{\mathbb{P}(\boldsymbol{x})} \prod_{\bar{w} \in \mathscr{C}(\tau)} \frac{c_{\bar{w}}(k) \times \Gamma(\sum_{u \in \mathscr{A}} \alpha_{\bar{w}u})}{k \times \Gamma(\sum_{u \in \mathscr{A}} N(\bar{w}u) + \alpha_{\bar{w},u})} \prod_{u \in \mathscr{A}} \frac{\Gamma(N(\bar{w}u) + \alpha_{\bar{w},u})}{\Gamma(\alpha_{\bar{w},u})} \tag{46}$$

Le terme $\mathbb{P}(\boldsymbol{x})$ est ici un terme de normalisation dont on peut éviter le calcul dans le cadre de la sélection de modèles. Cependant, nous verrons par la suite que son calcul est nécessaire dans le cadre de la classification bayésienne de séquences. Ce calcul est en fait tout à fait similaire à celui du calcul de la constante de normalisation $C(k)$ de la distribution a priori sur les modèles : il pose principalement la difficulté de requérir le calcul d'une somme sur les arbres de quantités définies par des produits sur les feuilles de chaque arbre. Nous verrons dans la partie suivante un algorithme permettant le calcul de ces constantes de normalisation.

Par ailleurs, il est possible de décomposer cette constante entre les feuilles, comme nous l'avons fait pour la distribution a priori, afin de retrouver un distribution exprimée comme un produit de quantités définies pour chaque feuille. En effet, notant $C(k, \boldsymbol{x}) = \sum_{\tau \in \mathscr{T}_l} \mathbb{P}(\tau, \boldsymbol{x})$, on peut alors définir :

$$c_{\bar{w}}(k, \boldsymbol{x}) = C(k, \boldsymbol{x})^{\frac{|\bar{w}|}{\mathscr{A}^l}}$$

ce qui permet d'écrire :

$$\mathbb{P}(\tau | \boldsymbol{x}) = \prod_{\bar{w} \in \mathscr{C}(\tau)} \frac{c_{\bar{w}}(k, \boldsymbol{x}) \Gamma(\sum_{u \in \mathscr{A}} \alpha_{\bar{w},u})}{k \Gamma(\sum_{u \in \mathscr{A}} N(\bar{w}u) + \alpha_{\bar{w},u})} \prod_{u \in \mathscr{A}} \frac{\Gamma(N(\bar{w}u) + \alpha_{\bar{w},u})}{\Gamma(\alpha_{\bar{w},u})}$$

2.6. Indépendance conditionnelle des partitions.

2.6.1. *Distribution a posteriori sur les partitions.*

Présence d'un nœud. Conditionnellement à une séquence observée $\boldsymbol{x} = (x_1, \ldots, x_n)$, les arbres sont distribués selon la probabilité a posteriori explicitée dans l'équation (46) du chapitre précédent. Par conséquent, si l'on note $\bar{\mathscr{C}}$ l'ensemble des motifs de longueur inférieure à l (*i.e.* $\bar{\mathscr{C}} = \cup_{i=1}^{l} \mathscr{C}_i$), si l'on définit, pour chaque nœud possible $\bar{\boldsymbol{w}} \in \bar{\mathscr{C}}$, l'événement de *présence du nœud* $\bar{\boldsymbol{w}}$ *dans l'arbre aléatoire* τ :

$$\mathscr{E}_{\bar{\boldsymbol{w}}} = \{\tau \in \mathscr{T}_l, \bar{\boldsymbol{w}} \in \bar{\mathscr{C}}(\tau)\}$$

En combinant ces événéments, on définit l'événement $\mathscr{P}_{\bar{\boldsymbol{w}}}(\bar{u}_1, \ldots, \bar{u}_p)$ de *présence de la partition* $(\bar{u}_1, \ldots, \bar{u_p})$ *sous le nœud* $\bar{\boldsymbol{w}}$ *de l'arbre* :

$$\mathscr{P}_{\bar{\boldsymbol{w}}}(\bar{u}_1, \ldots, \bar{u}_p) = \mathscr{E}_{\bar{\boldsymbol{w}}} \cap \mathscr{E}_{\bar{u}_1 \bar{\boldsymbol{w}}} \cap \cdots \cap \mathscr{E}_{\bar{u}_p \bar{\boldsymbol{w}}}$$

Frontière d'un arbre. Considérons à présent un arbre partiellement construit : ses branches se terminent par des feuilles, dont les labels $\bar{\boldsymbol{w}}_1, \ldots, \bar{\boldsymbol{w}}_q$ sont de longueur inférieure ou égale à l. Cependant, on impose que sous un nœud interne de l'arbre partiel, il y ait une partition complète de l'alphabet.

Les feuilles $\bar{\boldsymbol{w}}_1, \ldots, \bar{\boldsymbol{w}}_q$ d'un tel arbre forment une *frontière* de l'arbre. Conditionnellement à la présence d'une telle frontière dont toutes les feuilles sont de profondeur strictement inférieures à l, la probabilité que l'arbre se complète par les partitions $(\bar{u}_1^1, \ldots, \bar{u}_1^{p_1}), \ldots, (\bar{u}_q^1, \ldots, \bar{u}_q^{p_q})$ s'écrit :

$$\mathbb{P}(\cap_{i=1}^{q} \mathbb{P}_{\bar{\boldsymbol{w}}_i}(u_i^1, \ldots, u_i^{p_i}) | \boldsymbol{x}, \cap_{i=1}^{q} \mathscr{E}_{\bar{\boldsymbol{w}}_i}) = \frac{\sum_{\tau \in \cap_{i=1}^{q} \mathscr{P}_{\bar{\boldsymbol{w}}_i}(u_i^1, \ldots, u_i^{p_i})} \mathbb{P}(\tau | \boldsymbol{x})}{\sum_{\tau \in \cap_{i=1}^{q} \mathscr{E}_{\bar{\boldsymbol{w}}_i}} \mathbb{P}(\tau | \boldsymbol{x})} \tag{47}$$

2.6.2. *Indépendances conditionnelles.* Considérons à présent l'ensemble $\mathscr{T}_l(\bar{\boldsymbol{w}})$ des sous-arbres sous le nœud $\bar{\boldsymbol{w}}$.

Dans la relation précédente, chaque terme $\mathbb{P}(\tau|\boldsymbol{x})$ s'écrit comme un produit de quantités $V(\bar{\boldsymbol{w}})$ associées à chaque feuille $\bar{\boldsymbol{w}}$ de l'arbre τ :

$$\mathbb{P}(\tau | \boldsymbol{x}) = \prod_{\bar{\boldsymbol{w}} \in \bar{\mathscr{C}}(\tau)} \underbrace{\frac{c_{\bar{\boldsymbol{w}}} \times \Gamma(\sum_{u \in \mathscr{A}} \alpha_{\bar{\boldsymbol{w}}, u})}{k \times \Gamma(\sum_{u \in \mathscr{A}} N(\bar{\boldsymbol{w}} u) + \alpha_{\bar{\boldsymbol{w}}, u})}}_{V(\bar{\boldsymbol{w}})}$$

La probabilité des partitions (47) s'écrit alors :

$$\mathbb{P}(\cap_{i=1}^{q} \mathscr{P}_{\bar{\boldsymbol{w}}_i}(u_i^1, \ldots, u_i^{p_i}) | \boldsymbol{x}, \cap_{i=1}^{q} \mathscr{E}_{\bar{\boldsymbol{w}}_i}) = \frac{\sum_{\tau_1^1 \in \mathscr{T}_l(\bar{u}_1^1 \bar{\boldsymbol{w}}_1), \ldots, \tau_q^{p_q} \in \mathscr{T}_l(\bar{u}_q^{p_q} \bar{\boldsymbol{w}}_q)} \prod_{i=1}^{q} \prod_{j=1}^{p_i} \prod_{\bar{\boldsymbol{w}}' \in \bar{\mathscr{C}}(\tau_i^j)} V(\bar{\boldsymbol{w}}')}{\sum_{\tau_1 \in \mathscr{T}_l(\bar{\boldsymbol{w}}_1), \ldots, \tau_q \in \mathscr{T}_l(\bar{\boldsymbol{w}}_q)} \prod_{i=1}^{q} \prod_{\bar{\boldsymbol{w}}' \in \bar{\mathscr{C}}(\tau_i)} V(\bar{\boldsymbol{w}}')}$$

Or, pour chacune des sommes du numérateur et du dénominateur, la relation suivante intervient :

$$\sum_{\tau_1^1 \in \mathscr{T}_l(\bar{u}_1^1 \bar{\boldsymbol{w}}_1), \ldots, \tau_q^{p_q} \in \mathscr{T}_l(\bar{u}_q^{p_q} \bar{\boldsymbol{w}}_q)} \prod_{i=1}^{q} \prod_{j=1}^{p_i} \prod_{\bar{\boldsymbol{w}}' \in \bar{\mathscr{C}}(\tau_i^j)} V(\bar{\boldsymbol{w}}') = \prod_{i=1}^{q} \prod_{j=1}^{p_i} \sum_{\tau_i^j \in \mathscr{T}_l(\bar{u}_i^j \bar{\boldsymbol{w}}_i)} \prod_{\bar{\boldsymbol{w}}' \in \bar{\mathscr{C}}(\tau_i^j)} V(\bar{\boldsymbol{w}}') \tag{48}$$

ce qui permet la factorisation suivante de la probabilité des q partitions :

$$\mathbb{P}(\cap_{i=1}^{q} \mathbb{P}_{\bar{\boldsymbol{w}}_i}(u_i^1, \ldots, u_i^{p_i}) | \boldsymbol{x}, \cap_{i=1}^{q} \mathscr{E}_{\bar{\boldsymbol{w}}_i}) = \prod_{i=1}^{q} \frac{\prod_{j=1}^{p_i} \sum_{\tau_i^j \in \mathscr{T}(\bar{u}_i^j \bar{\boldsymbol{w}}_i)} V(\tau_i^j)}{\sum_{\tau_i \in \mathscr{T}(\bar{\boldsymbol{w}}_i)} V(\tau_i)} \tag{49}$$

avec $V(\tau_i) = \prod_{\bar{\boldsymbol{w}}' \in \bar{\mathscr{C}}(\tau_i)} V(\bar{\boldsymbol{w}}')$.

Par conséquent, la distribution sur les q partitions sous les nœuds formant la frontière s'écrit comme un produit de probabilités sur ces q partitions. Elles sont donc indépendantes sous la loi a posteriori.

Cette indépendance va intervenir à de nombreuses reprises. Elle est au cœur de l'algorithme de tirage aléatoire dans la distribution a posteriori sur les modèles, car elle assure que le tirage récursif de l'arbre est insensible à l'ordre dans lequel sont tirées les

partitions sous les nœuds. Par ailleurs, ce résultat montre que la distribution a posteriori des partitions sous un nœud de l'arbre ne dépend que des scores $V(\bar{w})$ des feuilles $\bar{w}$ possibles sous ce nœud.

CHAPITRE 7

Sélection de modèle

Nous dérivons à présent un critère de sélection de modèle à partir du formalisme bayésien établit précédemment. Ce critère de sélection de modèle est un critère *classique*, au sens où nous ne menons pas une estimation de la distribution sur les modèles. Pour une séquence observée $\boldsymbol{x} = (x_1, \ldots, x_n)$, le critère que nous proposons sélectionne un modèle comme le meilleur modèle pour décrire ces données.

1. Critère MAP

Nous allons naturellement dériver cet *estimateur* du modèle de la distribution a posteriori des modèles sachant la séquence observée $\boldsymbol{x}$. Le choix du *maximum a posteriori* s'impose : on retient le modèle que les données *supportent* le plus. Aussi, le critère que nous proposons sélectionne l'arbre $\hat{\tau}$ qui vérifie :

$$\hat{\tau} = \text{argmax}_{\tau \in \mathcal{T}_l} \mathbb{P}(\tau | \boldsymbol{x})$$

Ce critère possède deux qualités qui le rendent particulièrement puissant :

- compte-tenu des relations d'indépendances conditionnelles entre les partitions de l'arbre, il permet une maximisation exacte par un algorithme efficace de programmation dynamique
- asymptotiquement, il sélectionne le modèle de dimension la plus petite contenant la distribution ayant généré les données $\boldsymbol{x}$. Cela constitue un résultat de consistance au sens classique du critère de sélection de modèle, qui par ailleurs implique la dégénérescence asymptotique de la distribution a posteriori sur les modèles en une distribution de Dirac.

2. Consistance de la sélection de modèle

Considérons que la séquence $\boldsymbol{x}$ est générée selon une chaîne de Markov d'ordre l de paramètre $\theta^\star$, et que $\tau^\star$ est le modèle de Markov parcimonieux de plus petite dimension tel que $\theta^\star \in \Theta_{\tau^\star}$. Le résultat de consistance de la sélection de modèle s'appuie sur le résultat asymptotique suivant :

THÉORÈME 28. *Quelle que soit la constante de pénalité k sur les modèles, et pour une dsitribution a priori uniforme sur le paramètre de la chaîne de Markov d'ordre l, on a :*

$$\mathbb{P}(\tau^\star | \boldsymbol{x}) \to 1$$

Ce qui entraîne le corollaire suivant.

COROLLAIRE 28.1. *Dans les mêmes conditions que précédemment, on a :*

$$\mathbb{P}(\hat{\tau} = \tau^\star | \boldsymbol{x}) \to 1$$

2.1. Approximation de la distribution a posteriori des modèles. Ces résultats s'appuient sur le comportement de l'intégrale de la vraisemblance sous un modèle τ lorsque la longueur de la séquence tend vers l'infini. Afin de pouvoir mener cette approximation, nous factorisons le maximum de vraisemblance dans le modèle d'intérêt τ d'une manière différente selon que le modèle contient la vraie distribution $\boldsymbol{\theta}^\star$, ou non. Nous travaillons ici sur la distribution $\mathbb{P}(\tau, \boldsymbol{x})$, ce qui allège les notations.

Dans le premier cas, $\boldsymbol{\theta}^\star \in \Theta_\tau$, nous factorisons le maximum de vraisemblance $\hat{\boldsymbol{\theta}}_\tau$ dans le modèle τ :

$$\mathbb{P}(\tau, \boldsymbol{x}) = \frac{C(k)L(\hat{\boldsymbol{\theta}}_\tau, \boldsymbol{x})}{k^{|\mathscr{C}(\tau)|}} \int_{\Theta_\tau} \frac{L(\boldsymbol{\theta}_\tau, \boldsymbol{x})}{L(\hat{\boldsymbol{\theta}}_\tau, \boldsymbol{x})} p_\alpha(\boldsymbol{\theta}_\tau) \mathrm{d}\boldsymbol{\theta}_\tau$$

et effectuons le changement de variable $t_{\bar{\boldsymbol{w}},u} = \frac{\theta_{\bar{\boldsymbol{w}},u} - \hat{\theta}_{\bar{\boldsymbol{w}},u}}{\hat{\theta}_{\bar{\boldsymbol{w}},u}}$. On remarque alors que la relation suivante lie les paramètres du domaine de la nouvelle variable $\boldsymbol{u}$:

$$\forall \bar{\boldsymbol{w}} \in \mathscr{C}(\tau), \langle \hat{\boldsymbol{\theta}}_{\bar{\boldsymbol{w}},u}, \boldsymbol{t}_{\bar{\boldsymbol{w}},u} \rangle = 0$$

Notant :

$$\Theta'_\tau = \left\{ \boldsymbol{t} \in \prod_{\bar{\boldsymbol{w}} \in \mathscr{C}(\tau), u \in \mathscr{A}}]-1, \frac{1-\hat{\theta}_{\bar{\boldsymbol{w}},u}}{\hat{\theta}_{\bar{\boldsymbol{w}},u}}[, \forall \bar{\boldsymbol{w}} \in \mathscr{C}(\tau), \langle \hat{\boldsymbol{\theta}}_{\bar{\boldsymbol{w}}}, \boldsymbol{t}_{\bar{\boldsymbol{w}}} \rangle = 0 \right\}$$

il vient alors :

$$\mathbb{P}(\tau, \boldsymbol{x}) = \mathbb{P}_{\hat{\theta}}(\boldsymbol{x}) \times \int_{\Theta'_\tau} \exp \tilde{n} \sum_{\bar{\boldsymbol{w}} \in \mathscr{C}(\tau)} \frac{\tilde{N}(\bar{\boldsymbol{w}})}{\tilde{n}} \sum_{u \in \mathscr{A}} \frac{\tilde{N}(\bar{\boldsymbol{w}}, u)}{\tilde{N}(\bar{\boldsymbol{w}})} (\log(1 + t_{\bar{\boldsymbol{w}},u}) - t_{\bar{\boldsymbol{w}},u}) \mathrm{d}\boldsymbol{u}$$

où $\tilde{N}(\bar{\boldsymbol{w}}u) = N(\bar{\boldsymbol{w}}u) + \alpha_{\bar{\boldsymbol{w}},u} - 1$, $\tilde{N}(\bar{\boldsymbol{w}}) = N(\bar{\boldsymbol{w}}) + \sum_{u \in \mathscr{A}} \alpha_{\bar{\boldsymbol{w}},u} - 1$, et $\tilde{n} = n + \sum_{(\bar{\boldsymbol{w}},u) \in \mathscr{C}(\tau) \times \mathscr{A}} \alpha_{\bar{\boldsymbol{w}},u} - 1$.

Compte-tenu qu'un modèle de Markov parcimonieux est un sous-modèle linéaire d'un modèle différentiable en moyenne quadratique, la consistance de l'estimateur du maximum de vraisemblance dans chaque modèle parcimonieux dont les transitions sont toutes strictement positives découle des résultats classiques (voir [**?**]). En l'occurence, nous nous intéressons surtout à l'estimateur MAP $\tilde{\boldsymbol{\theta}}_\tau$ défini par la relation :

$$\forall (\bar{\boldsymbol{w}}, u) \in \mathscr{C}(\tau) \times \mathscr{A}, \tilde{\theta}_{\bar{\boldsymbol{w}},u} = \frac{\tilde{N}(\bar{\boldsymbol{w}}u)}{\sum_{v \in \mathscr{A}} \tilde{N}(\bar{\boldsymbol{w}}v)}$$

Pour $\alpha > 0$ et des constantes f^+ et f^- strictement positives, on a donc l'encadrement suivant dès lors que n est suffisamment grand :

$$\mathbb{P}(\forall (\bar{\boldsymbol{w}}, u) \in \mathscr{C}(\tau) \times \mathscr{A}, f^- \leq \frac{N(\bar{\boldsymbol{w}}u)}{n} \leq f^+) > 1 - \alpha$$

d'où, compte-tenu de l'égalité entre $\tilde{\boldsymbol{\theta}}$ et $\hat{\boldsymbol{\theta}}$ dans le cas du prior uniforme :

$$\mathbb{P}(\forall (\bar{\boldsymbol{w}}, u) \in \mathscr{C}(\tau) \times \mathscr{A}, f^- \leq \frac{\tilde{N}(\bar{\boldsymbol{w}}u)}{\tilde{n}} \leq f^+) > 1 - \alpha$$

pour n suffisamment grand.

Dès lors, on peut établir l'encadrement suivant pour la probabilité jointe du modèle τ et de la séquence $\boldsymbol{x}$:

$$\int_{\Theta'_\tau} \exp n f^+ \sum_{(\bar{\boldsymbol{w}},u) \in \mathscr{C}(\tau) \times \mathscr{A}} \log(1 + t_{\bar{\boldsymbol{w}},u}) - t_{\bar{\boldsymbol{w}},u} \, \mathrm{d}\boldsymbol{t} \leq \frac{\mathbb{P}(\tau, \boldsymbol{x})}{\mathbb{P}_{\hat{\boldsymbol{\theta}}_\tau}(\boldsymbol{x})} \leq \int_{\Theta'_\tau} \exp n f^- \sum_{(\bar{\boldsymbol{w}},u) \in \mathscr{C}(\tau) \times \mathscr{A}} \log(1 + t_{\bar{\boldsymbol{w}},u}) - t_{\bar{\boldsymbol{w}},u} \, \mathrm{d}\boldsymbol{t} \tag{50}$$

valable avec probabilité $1 - \alpha$ sous la distribution de la source $\mathbb{P}_{\boldsymbol{\theta}^\star}$, dès lors que la longueur n de la séquence $\boldsymbol{x}$ est suffisamment grande.

Pour le cas où $\boldsymbol{\theta}^\star \in \Theta_l \backslash \Theta_\tau$, on utilise une technique similaire, mais en factorisant le maximum de vraisemblance $\hat{\theta}$ dans le modèle de Markov d'ordre l moyennant un plongement du paramètre dans le modèle τ. Cela conduit à l'encadrement suivant :

$$\frac{\mathbb{P}(\tau, \boldsymbol{x})}{\mathbb{P}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})} \leq \int_{\Theta'_\tau} \exp n f^- \sum_{(\boldsymbol{w},u) \in \mathscr{A}^{l+1}} \log(1 + t_{\boldsymbol{w},u}) - t_{\boldsymbol{w},u} \, \mathrm{d}\boldsymbol{t} \tag{51}$$

La différence essentielle dans ce cas est que l'ensemble Θ_τ étant inclus dans un sous-espace des paramètres qui ne contient pas la véritable distribution $\boldsymbol{\theta}^\star$, il ne contient pas non plus l'estimateur du maximum de vraisemblance $\hat{\boldsymbol{\theta}}$, du moins avec une probabilité

supérieure à $1-\alpha$ dès lors que la séquence est suffisamment longue. Par conséquent, conditionnellement à l'évenement $\hat{\boldsymbol{\theta}} \notin \Theta_\tau$, il existe $\varepsilon > 0$ tel que $B_\infty(\mathbf{0},\varepsilon) \cap \Theta'_\tau = \emptyset$. Cette différence est à l'origine de la différence de comportement asymptotique entre les probabilités d'arbres contenant la distribution des données et de ceux ne la contenant pas.

2.2. Expansion de Laplace. La description du comportement des intégrales obtenues précédemment est décrit par une technique d'analyse appelée *l'expansion de Laplace*. Elle permet d'établir le résultat suivant :

LEMME 2. *Soit $\mathscr{D}$ un domaine borné d'un sous-espace affine de dimension d de $\mathbb{R}^k$, avec $d < k$, et une constante $C > 0$. Alors :*

$$\int_{\mathscr{D}} \exp nC \sum_{i=1}^{k} (\log(1+t_i) - t_i) \mathrm{d}\boldsymbol{t} = \begin{cases} \phi_n \sqrt{n}^{-d} & si\ 0 \in D \\ \phi'_n \exp -nC\delta & si \exists \varepsilon > 0, \mathscr{D} \cap B_\infty(\mathbf{0},\varepsilon) = \emptyset \end{cases}$$

avec $0 < a < \phi_n < b$ et $\phi_n{\prime} < b$, a et b étant des constantes positives.

Ce résultat s'appuie sur les propriétés de la fonction $u \mapsto \log(1+u) - u$ admet un maximum unique en $\boldsymbol{u} = 0$. Par conséquent, dès lors que $\mathbf{0}$ est à une distance d'au moins ε de $\mathscr{D}$, il suffit de poser $\delta(\varepsilon) = -\max(\log(1+\varepsilon) - \varepsilon, \log(1-\varepsilon) + \varepsilon)$ pour avoir :

$$\forall \boldsymbol{t} \in \mathscr{D}, \exp nC \sum_{i=1}^{k} \log(1+t_i) - t_i < \exp -nC\delta(\varepsilon)$$

Remarquant que $\delta(\varepsilon) > 0$ dès lors que $\varepsilon > 0$, il vient :

$$\phi'_n = \frac{1}{\exp -nC\delta(\varepsilon)} \int_{\mathscr{D}} \exp nC \sum_{i=1}^{k} \log(1+t_i) - t_i \, \mathrm{d}\boldsymbol{t} < \int_{\mathscr{D}} \mathrm{d}\boldsymbol{t}$$

Dans le cas où $\mathbf{0}$ appartient au domaine, le calcul est plus délicat. Partageons l'ensemble $\mathscr{D}$ en deux parties disjointes :

– l'intersection avec $B_\infty(\mathbf{0},\varepsilon)$, notée $\mathscr{D}_+ = \mathscr{D} \cap B_\infty(\mathbf{0},\varepsilon)$,

– son complémentaire, noté $\mathscr{D}_- = \mathscr{D} \setminus B_\infty(\mathbf{0},\varepsilon)$.

Sur $\mathscr{D}_+$, le développement en série entière suivant est valable, dès lors que $\varepsilon < 1$:

$$\forall i \in \{1,\ldots,k\}, \log(1+t_i) - t_i = -\frac{t_i^2}{2} + \sum_{j=1}^{\infty} \frac{(-t_i)^j}{j}$$

Or,

$$\forall t_i \in]-\varepsilon,\varepsilon[, \min(-\frac{t_i}{3}; 0) \leq \sum_{j=3}^{\infty} \frac{(-t_i)^{j-2}}{j} \leq \max(-\frac{t_i}{3} + \frac{t_i^2}{4}, \frac{-t_i}{1+t_i})$$

d'où l'encadrement :

$$1 - \frac{\varepsilon}{3} \leq 1 + 2 \sum_{j=3}^{\infty} \frac{(-t_i)^{j-2}}{j} \leq 1 + \frac{\varepsilon}{1-\varepsilon}$$

Le report de ces encadrements dans l'intégrale montre que :

$$\begin{aligned} (52) \qquad & \int_{\mathscr{D}^+} \exp -n \frac{C(1+\varepsilon)}{2(1-\varepsilon)} \sum_{i=1}^{k} t_i^2 \mathrm{d}\boldsymbol{t} \\ & \leq \int_{\mathscr{D}^+} \exp -nC \sum_{i=1}^{k} k \log(1+t_i) - t_i \\ & \leq \int_{\mathscr{D}_+} \exp -nC(1-\frac{2\varepsilon}{3}) \sum_{i=1}^{k} t_i^2 \mathrm{d}\boldsymbol{t} \end{aligned}$$

Le changement de variable $r_i = \sqrt{n} t_i$ permet permet alors d'écrire pour $A > 0$:

$$\sqrt{n}^{-d} \int_{B(\mathbf{0},1} \exp -nA \sum_{i=1}^{k} r_i^2 \mathrm{d}\boldsymbol{r} \leq \int_{\mathscr{D}_+} \exp -nA \sum_{i=1}^{k} t_i^2 \mathrm{d}\boldsymbol{t} \leq \sqrt{n}^{-d} \int_{\mathbb{R}^d} \exp -A \sum_{i=1}^{k} r_i^2 \mathrm{d}\boldsymbol{r}$$

car, pour n suffisamment grand, $\sqrt{n}\mathscr{D}_+$ contient la boule unité de $\mathbb{R}^d$, et est contenu dans $\mathbb{R}^d$. Par ailleurs, le terme en $\sqrt{n}^d$ provient du changement de la forme différentielle associée au changement de variable. On en tire finalement que :

$$\int_{B(\mathbf{0},1} \exp -\frac{C}{2}\sum_{i=1}^{k} r_i^2 \mathrm{d}\boldsymbol{r} \leq \phi_n \leq \int_{\mathbb{R}^d} \exp -\frac{C}{3}\sum_{i=1}^{k} r_i^2 \mathrm{d}\boldsymbol{r} + \sqrt{n}^d \exp -nC\delta(\varepsilon)$$

ce qui prouve le résultat. □

2.3. Conclusion. Nous exploitons à présent le lemme 7 pour obtenir des encadrements explicites à partir des équations 50 et 51. L'équation 50 permet en effet d'établir la relation :

$$\phi_n^+ \sqrt{n}^{-d} \leq \frac{\mathbb{P}(\tau, \boldsymbol{x})}{\mathbb{P}_{\hat{\boldsymbol{\theta}}_\tau}(\boldsymbol{x})} \leq \phi_n^- \sqrt{n}^{-d} \tag{53}$$

qui est valable avec probabilité au moins $1-\alpha$ sous $\mathbb{P}_{\boldsymbol{\theta}^\star}$ pour n suffisamment grand. De la même manière, on dérive de la relation 51 la majoration suivante :

$$\frac{\mathbb{P}(\tau, \boldsymbol{x})}{\mathbb{P}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})} \leq \phi'_n \exp -nf^-\delta(\varepsilon) \tag{54}$$

Le rapport des probabilités a posteriori du modèle $\tau^\star$ et d'un modèle τ suit donc une asymptotique distincte selon que le modèle τ contient la distribution $\boldsymbol{\theta}^\star$ ou non :

$$\log \frac{\mathbb{P}(\tau^\star|\boldsymbol{x})}{\mathbb{P}(\tau|\boldsymbol{x})} \geq \begin{cases} \log \frac{\mathbb{P}_{\hat{\theta}_{\tau^\star}}(\boldsymbol{x})}{\mathbb{P}_{\hat{\theta}_\tau}(\boldsymbol{x})} + \delta \log \phi_n + \frac{(|\mathscr{A}|-1)(|\mathscr{C}(\tau)|-|\mathscr{C}(\tau^\star)|}{2} \log n & \text{si } \theta^\star \in \Theta_\tau \\ \log \frac{\mathbb{P}_{\hat{\theta}_{\tau^\star}}(\boldsymbol{x})}{\mathbb{P}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})} + \log \phi_n^+ - \frac{(|\mathscr{A}|-1)|\mathscr{C}(\tau^\star)|}{2} \log n + Cn\delta(\varepsilon) & \text{sinon} \end{cases} \tag{55}$$

avec $\delta \log \phi_n$ égal à la différence des logarithmes des termes ϕ_n^+ et ϕ_n^- obtenus dans chacun des modèles $\tau^\star$ et τ respectivement, d'après le lemme 7. Ces relations ne sont valables que dans l'asymptotique, avec une probabilité supérieure à $1-\alpha$ sous la distribution des données $\mathbb{P}^\star_{\boldsymbol{\theta}}$, pour une longueur n de la séquence $\boldsymbol{x}$ suffisamment grande. Dans les deux cas, le premier terme est une statistique de test de rapport de vraisemblance entre modèles emboités, dont la distribution est bornée avec probabilité au moins $1-\alpha$. Il est alors immédiat de remarquer que la différence des logarithmes des probabilités a posteriori tend vers l'infini dans le cas où $\boldsymbol{\theta}^\star \notin \Theta_\tau$. Dans le premier cas, si le modèle τ contient la distribution des données mais n'est pas le modèle minimal $\tau^\star$, alors $|\mathscr{C}(\tau)| > |\mathscr{C}(\tau^\star)|$. Le terme en $\log n$ est donc le terme dominant, et le rapport des probabilités a posteriori croît également vers l'infini.

On montre ainsi que pour tout $\alpha > 0$, il existe une longueur de séquence à partir de laquelle :

$$\mathbb{P}(\tau^\star|\boldsymbol{x}) > \mathbb{P}(\tau|\boldsymbol{x})$$

pour tout autre modèle τ, avec une probabilité supérieure à $1-\alpha$ sous $\mathbb{P}_{\boldsymbol{\theta}^\star}$. □

2.4. Comparaison avec le critère BIC. Le résultat précédent montre que la probabilité a posteriori d'un modèle τ :

- décroît strictement plus vite que le critère BIC si τ contient des contraintes que la matrice θ ayant généré les données ne vérifie pas,
- est asymptotiquement équivalent au critère BIC si τ a pu générer les données.

Ce résultat montre que le critère bayésien est théoriquement plus efficace, dans la mesure où il disqualifie les modèles ne contenant pas la loi des données plus rapidement.

Par ailleurs, l'équivalent que nous établissons n'implique pas que le critère bayésien classe asymptotiquement les modèles dans le même ordre que l critère BIC. L'équivalent établi est en effet à l'ordre $\log n$, et rien n'interdit a priori que l'écart entre le BIC de deux modèles reste borné à un ordre inférieur. Cependant, le résultat entraîne que les

modèles sélectionnés par les deux critères coïncident asymptotiquement avec une probabilité 1 sous les hypothèses classiques. Le résultat de consistance est par conséquent également valide pour le critère BIC.

Quatrième partie

Modèles de Markov parcimonieux, algorithmes et applications

Introduction

Le résultat de consistance de la sélection bayésienne d'un modèle parcimonieux étant acquis, nous abordons à présent la question de sa mise en œuvre. Il s'agit de construire des algorithmes permettant, d'une part, la recherche de l'arbre maximisant la probabilité de la séquence (une sorte de CTM parcimonieux), et d'autre part, l'évaluation de la probabilité de la séquence sous le modèle bayésien, c'est-à-dire en aggrégeant l'ensemble des modèles (un CTW parcimonieux).

Dans le cas des arbres de contextes, ces tâches sont réalisées au moyen des deux algorithmes classiques, CTM et CTW respectivement, dont les caractéristiques en termes de complexité algorithmique sont enviables : ces algorithmes permettent en effet de réaliser ces deux tâches avec une complexité algorithmique linéaire en fonction de la longueur de la séquence. Ils présentent une seconde caractéristique attrayante, à savoir la possibilité de recalculer la distribution de chaque symbole en chaque position. Ce dernier aspect est essentiel pour les applications à la compression, mais d'un importance limitée pour les applications statistiques typiques à l'analyse de séquences biologiques.

Les algorithmes que nous proposons généralisent CTM et CTW au cas des modèles parcimonieux. Bien que développés indépendamment, alors que l'auteur ne disposait d'aucune connaissance de ces algorithmes, les algorithmes PCTM et PCTW (pour Parsimonious Context Tree Maximization *et* Parsimonious Context Tree Weighting *respectivement) exploitent le même principe de programmation dynamique. De même, une méthode de détermination d'un ensemble de modèles candidats adéquat est utilisée ; elle permet de ne pas considérer des modèles a priori indistinguables, et par conséquent de réduire la complexité des algorithmes.*

Enfin, des évaluations des performances des modèles parcimonieux pour la modélisation de séquences codantes issues de génomes bactériens sont présentées, s'appuyant sur une évaluation de la qualité d'ajustement des modèles sélectionnés aux séquences en comparaison à des modèles de MARKOV *d'ordre fixe. Une évaluation comparative des performances de ces modèles comparé à l'ensemble du spectre des modèles pour séquences (chaînes de* MARKOV *d'ordre fixe, variable, réseaux bayésiens) fait l'objet d'une collaboration en cours avec l'équipe de bioinformatique de l'*Institut für Pflanzengenetik und Kulturpflanzenforschung *(IPK). Le champ d'application exploré dans ce dernier travail est la détection de sites de fixation de facteurs de transcription dans les génomes microbiens.*

CHAPITRE 8

Algorithme exact de sélection du MAP

1. Programmation dynamique

Compte-tenu du nombre de modèles parcimonieux d'ordre l présenté dans le tableau 3, il n'est pas envisageable de maximiser le critère de sélection de modèle d'une manière gloutonne. Il est donc nécessaire de construire un algorithme récursif, tel que *context* dans le cas des chaînes de Markov à longueur variable, qui réalise cette maximisation.

A cette fin, nous montrons que le problème de maximisation posé est compatible avec une relation de récurrence, qui permet sa résolution par un algorithme de programmation dynamique.

1.1. Notations. Nous adoptons ici des notations classiques d'optimisation. Rappelons que si $\boldsymbol{x}$ désigne la séquence observée, le critère à maximiser est de la forme :

$$\mathbb{P}(\tau|\boldsymbol{x}) = \prod_{\bar{\boldsymbol{w}}\in\mathscr{C}(\tau)} V(\bar{\boldsymbol{w}})$$

En échelle logarithmique, le problème se reformule aisément sous la forme :

$$\log\mathbb{P}(\tau|\boldsymbol{x}) = \sum_{\bar{\boldsymbol{w}}\in\mathscr{C}(\tau)} v(\bar{\boldsymbol{w}})$$

où $v(\bar{\boldsymbol{w}}) = \log V(\bar{\boldsymbol{w}})$. Notons $\hat{\tau}$ l'arbre qui maximise le critère, et $\hat{\tau}(\bar{\boldsymbol{w}})$ le sous-arbre sous $\bar{\boldsymbol{w}}$ dont la somme des scores v des feuilles est maximale :

$$\hat{\tau}(\bar{\boldsymbol{w}}) = \underset{\tau(\bar{\boldsymbol{w}})\in\mathscr{T}_l(\bar{\boldsymbol{w}})}{\operatorname{argmax}} \underbrace{\sum_{\bar{\boldsymbol{w}}'\in\mathscr{C}(\tau(\bar{\boldsymbol{w}}))} v(\bar{\boldsymbol{w}}')}_{v(\tau(\bar{\boldsymbol{w}}))}$$

1.2. Relation de récurrence. Soit $\bar{w}$ un motif de $\bar{\mathscr{C}_l}$. Le sous-arbre $\tau^\star(\bar{\boldsymbol{w}})$ sous ce nœud qui maximise la somme partielle $v(\tau(\bar{\boldsymbol{w}})$ se déduit des arbres $\tau^\star(\bar{u}\bar{\boldsymbol{w}})$, pour tout symbole $\bar{u}\in\bar{\mathscr{A}}$ de la manière suivante.

THÉORÈME 29. *Le sous-arbre optimal $\tau^\star(\bar{\boldsymbol{w}})$ sous un nœud $\bar{\boldsymbol{w}}\in\bar{\mathscr{C}_l}$ vérifie :*

$$\tau^\star(\bar{\boldsymbol{w}}) = \underset{\bar{u}_1,\ldots,\bar{u}_p\ partition\ de\ \bar{\mathscr{A}}}{\operatorname{argmax}} \sum_{i=1}^{p} v(\tau^\star(\bar{u}_i\bar{\boldsymbol{w}})) \tag{56}$$

Pour prouver le théorème 29, on remarque d'abord la propriété suivante, qui découle de la structure de somme sur les feuilles de l'arbre de la quantité à maximiser. En effet, quelque soit le choix de la partition $(\bar{u}_1,\ldots,\bar{u}_p)$ sous le nœud $\bar{\boldsymbol{w}}$, on a la relation :

$$\forall(\tau_1,\ldots,\tau_p)\in\mathscr{T}_l(\bar{u}_1\bar{\boldsymbol{w}})\times\cdots\times\mathscr{T}_l(\bar{u}_p\bar{\boldsymbol{w}}), v(\tau_1,\ldots,\tau_p) \leq \sum_{i=1}^{p} v(\tau^\star(\bar{u}_i\bar{\boldsymbol{w}}))$$

où l'on identifie au vecteur $(\tau_1,\ldots,\tau_p)$ l'arbre $\tau\in\mathscr{T}_l(\bar{\boldsymbol{w}})$ dont la partition sous le nœud $\bar{\boldsymbol{w}}$ est $(\bar{u}_1,\ldots,\bar{u}_p)$, et dont les sous arbres sous chacun de ces nœuds sont respectivement les $\tau_1,\ldots,\tau_p$. Comme, dans la relation précédente, l'égalité a lieu lorsque chacun des τ_i vaut précisément $\tau^\star(\bar{u}_i\bar{\boldsymbol{w}})$, le maximum du membre de gauche est obtenu en choisissant la partition $(\bar{u}_1,\ldots,\bar{u}_p)$ donnant lieu à la plus grande somme à droite. □

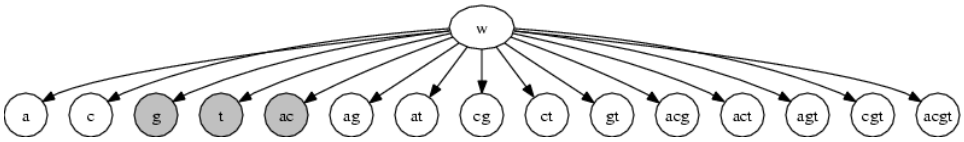


FIG. 1. Exemple d'ensemble de nœuds sous un nœud interne de l'arbre étendu. L'alphabet est celui des nucléotides $\{A, C, G, T\}$.

Ce résultat montre que le critère bayésien, global au modèle, prend également une forme locale dans l'arbre : son optimisation appelle en effet l'optimisation de chacun de ses sous-arbres sous le même critère. Par comparaison avec l'algorithme *context*, ce critère partage le fait d'admettre une implémentation récursive, mais présente l'avantage d'être mieux fondé statistiquement.

2. Algorithme de sélection de modèle

Nous explicitons à présent l'algorithme récursif qui implémente la récursion précédente. Pour cela, nous avons besoin d'un système de représentation de l'ensemble des arbres, dans lequel nous pourrons considérer l'ensemble des nœuds possibles dans l'arbre. Ce système de représentation est fourni par l'*arbre étendu*, que nous définissons dans le premier paragraphe. Le second paragraphe décrit le déroulement de l'algorithme dans cet arbre étendu.

2.1. Arbre étendu.

2.1.1. *Définition.* L'*arbre étendu* est l'arbre contenant l'ensemble des arbres d'une profondeur donné. L'ensemble des nœuds présents sous un nœud donné doit donc coïncider avec l'ensemble des symboles $\bar{\mathscr{A}}$. Un tel arbre est difficile à représenter graphiquement, compte-tenu du grand nombre de nœuds qui le compose, mais la figure 1 montre l'ensemble des nœuds sous un nœud interne $\bar{w}$ donné.

Contrairement au dénombrement des arbres parcimonieux, le dénombrement des nœuds de l'arbre étendu est immédiat : si $\mathscr{N}(|\mathscr{A}|, l)$ désigne le nombre de nœuds de l'arbre étendu pour un alphabet de cardinal $|\mathscr{A}|$ et un ordre l, on a la relation :

$$\mathscr{N}(|\mathscr{A}|, l) = \sum_{i=0}^{l} (2^{|\mathscr{A}|} - 1)^i$$

d'où :

$$\mathscr{N}(|\mathscr{A}|, l) = \frac{(2^{|\mathscr{A}|} - 1)^l - 1}{2^{|\mathscr{A}|} - 2}$$

Dans l'arbre étendu, il est alors possible de stocker les quantités $v(\tau^\star(\bar{w}))$ pour tous les nœuds possibles. C'est donc le cadre naturel dans lequel formuler l'algorithme de sélection de modèle.

2.2. Algorithme. L'algorithme de sélection de modèle s'écrit alors d'une manière très similaire à l'algorithme *context*. La seule différence, finalement, est que nous élaguons des partitions au lieu d'élaguer des nœuds à chaque pas.

Il suffit donc d'itérer la récursion suivante, depuis les feuilles jusqu'à la racine :

(1) pour toute feuille $\bar{w} \in \mathscr{C}_l$ de l'arbre étendu, calculer la quantité $v(\bar{w})$

(2) monter d'un niveau, et pour chaque nœud $\bar{w}$ du niveau courant :

   (a) pour chaque partition de l'alphabet $\bar{u}_1, \ldots, \bar{u}_p$ sous le nœud courant, calculer $\sum_{i=1}^{p} v(\tau^\star(\bar{u}_i \bar{w}))$,

   (b) éliminer les fils du nœud courant qui ne participent pas à la partition $\bar{u}_1, \ldots, \bar{u}_p$ réalisant le maximum, et stocker ce maximum $v(\tau^\star(\bar{w}))$

(3) retourner en 2 si la racine n'est pas atteinte.

A l'issue de cet algorithme, l'arbre obtenu est celui qui maximise le critère de sélection de modèle, en vertu du théorème 29.

3. Optimisation du BIC

Un autre critère populaire pour la sélection de modèles dont le nombre de paramètres est variable est le critère BIC. Comme nous l'avons rappelé précédemment, la consistance de ce critère pour la sélection de l'ordre d'une chaîne de Markov est un résultat aujourd'hui classique.

Un aspect attrayant de l'algorithme présenté précédemment est qu'il permet tout autant l'optimisation du critère BIC que celle du critère bayésien. En effet, le critère BIC d'adéquation d'un modèle τ à une séquence $\boldsymbol{x}$ s'écrit :

$$BIC(\tau, \boldsymbol{x}) = -\log L(\hat{\theta}_\tau) + \frac{|\mathscr{C}(\tau)|(|\mathscr{A}|-1)}{2}\log n$$

où le second terme est une pénalisation proportionnelle au nombre de paramètres du modèle et au logarithme de la longueur de la séquence. Or, cette expression se réécrit comme une somme sur les feuilles de l'arbre :

$$BIC(\tau, \boldsymbol{x}) = \sum_{\bar{w}\in\mathscr{C}(\tau)} (|\mathscr{A}|-1)\log n - N(\bar{w}u)\log\hat{\theta}_\tau(\bar{w}, u)$$

ce qui assure de pouvoir réaliser son optimisation (en l'occurence sa minimisation) par la même programmation dynamique que précédemment.

Notons pour conclure que cet algorithme est précisément la généralisation au cadre des modèles parcimonieux de l'algorithme développé par Zsolt Talata pour la sélection des VLMC par minisation du critère BIC.

4. Complexité algorithmique

Nous proposons à présent une évaluation de la complexité des algorithmes proposés. Il s'avère que les deux algorithmes présentent la même complexité, et que celle-ci peut-être aisément approchées grâce au formalisme de l'arbre étendu.

4.1. Remarques préliminaires. Trois remarques s'imposent avant d'envisager ce calcul. La première s'appuie sur la similarité des deux algorithmes présentés, qui ne diffèrent que par la substitution d'un opérateur *somme* à un opérateur *max*. Les arguments de chacun de ces opérateurs étant les mêmes dans les deux cas, ils présentent la même complexité si l'on accorde un coût égal aux opérateurs eux-mêmes.

La seconde remarque découle de la représentation des algorithmes dans l'arbre étendu. Dans chacun des deux algorithmes, la récurrence conduit à envisager chacun des nœuds internes de cet arbre. Et pour chacun de ces nœuds, les algorithmes requièrent d'effectuer autant de sommations que de partitions possibles sous l'arbre. Ainsi, hors l'évaluation de la complexité des calculs des scores des feuilles, la complexité des algorithmes se réduit donc à un produit du nombre de nœuds internes par le nombre d'opérations à effectuer pour chacun.

Dernier point : l'algorithme requiert, préalablement, le dénombrement des occurences de l'ensemble des mots de longueur l dans la séquence. Cette tâche peut-être réalisée en temps linéaire par rapport à la longueur de la séquence, en utilisant les arbres de suffixe par exemple. On note C_d la complexité de ce dénombrement.

4.2. Calcul de la complexité. Grâce au recours à l'arbre étendu, le calcul de la complexité de cet algorithme est assez simple. On remarque en effet au déroulement de l'algorithme qu'il suffit d'examiner chaque nœud interne de l'arbre, et pour chacun d'eux, d'envisager toutes les partitions possibles sous chacun.

4.2.1. *Dénombrement des feuilles de l'arbre étendu.* L'arbre étendu présente toujours le même nombre de nœuds enfants sous chaque nœud interne, nombre égal au cardinal de l'ensemble des parties non vides de l'alphabet. Ce nombre $P(\mathscr{A})$ est aisé à évaluer : il y a deux possibilités pour chaque élément, qui peut soit appartenir à la partie, soit ne pas lui appartenir. Soit :

$$P(\mathscr{A}) = 2^{|\mathscr{A}|} - 1 \tag{57}$$

compte-tenu qu'il faut retirer la partie vide.

Compte-tenu de la structure d'arbre, le nombre de nœuds $N(k)$ à la profondeur k répond à la relation :

$$N(k) = P(\mathscr{A})^k$$

et le nombre de feuilles N_f est donc :

$$N_f = P(\mathscr{A})^l$$

Pour chacune des feuilles, il faut évaluer $|\mathscr{A}|$ quantités distinctes afin d'évaluer le score $V(\bar{w})$. Par conséquent, le nombre d'opérations C_f nécessaire à l'évaluation de l'ensemble des scores des feuilles est de l'ordre :

$$C_f = O(P(\mathscr{A})^l \times |\mathscr{A}|)$$

4.2.2. *Dénombrement des nœuds internes.* Une fois les scores des feuilles évalués, les étapes successives de la récurrence impose de considérer exactement une fois chaque nœud interne. Le nombre de ces nœuds N_i se décompose comme la somme des nombres de nœuds à chaque niveau, ce qui conduit à l'équation suivante compte-tenu de 57 :

$$N_i = \sum_{k=0}^{l-1} P(\mathscr{A})^k = \frac{1 - P(\mathscr{A})^l}{1 - P(\mathscr{A})}$$

Pour chaque nœud interne, il faut alors considérer chacune des partitions. Leur nombre $\mathscr{N}(\mathscr{A})$ a été calculé dans la seconde partie (voir 37), et on rappelle que :

$$\mathscr{N}(\mathscr{A}) = \frac{1}{e} \sum_{k=1}^{\infty} \frac{k^{|\mathscr{A}|}}{k!}$$

Enfin, chacune de ces partitions voit son score évalué au moyen de la sommation d'un nombre de termes égal au nombre de parties de la partition, que l'on majore par le cardinal de l'alphabet.

Ainsi, la complexité C_i liée aux étapes de la récursion portant sur les nœuds internes s'écrit :

$$C_i = N_i \times \mathscr{N}(\mathscr{A}) \times |\mathscr{A}|$$

Par conséquent, la complexité C des algorithmes s'écrit :

$$C = C_d + C_f + C_i = O(2^{|\mathscr{A}| \times l} \times |\mathscr{A}|) + O(2^{|\mathscr{A}| \times l} \times \exp(|\mathscr{A}|) \times |\mathscr{A}|)$$

soit encore :

$$C = O(n) + O(\exp(|\mathscr{A}|(l+1))|\mathscr{A}|)$$

4.3. Remarques. La complexité C calculée précédemment présente une croissance linéaire par rapport à la longueur de la séquence, ce qui peut sembler raisonnable. Mais les autres termes ne sont pas à négliger, puisqu'ils induisent les limites informatiques en termes de taille d'alphabet.

En effet, la croissance en $\exp(|\mathscr{A}|(l+1))|\mathscr{A}|$ de C est extrêmement rapide. Cependant, elle reste petite devant le nombre des modèles, dont l'équivalent est de la forme :

$$\mathscr{N}(\mathscr{A}, l) = \exp(|\mathscr{A}|^l)$$

Cette différence drastique est le bénéfice résultant du recours à la programmation dynamique, sans laquelle ce critère ne serait simplement pas implémentable.

5. Calcul de la probabilité de la séquence

5.1. Sommer plutôt que maximiser. Nous avons jusqu'à présent travaillé avec une distribution sur les modèles normalisée de manière théorique par une constante C_k. Nous nous intéressons à présent au problème de son calcul.

L'algorithme permettant ce calcul est intimement lié au précédent. La différence tient uniquement au fait que nous étions intéressés par la maximisation du critère, et que nous cherchons à présent à le sommer sur les modèles. Nous allons voir à présent qu'il suffit de remplacer l'opérateur max par l'opérateur $\sum$ pour réaliser ce calcul. Ce faisant, on construit de nouveau un algorithme de programmation dynamique, qui permet de contourner l'écueil combinatoire lié à l'immense quantité de modèles envisagée.

5.2. Algorithme.

5.2.1. *Commutation des produits et des sommes.* Sous un nœud $\bar{w} \in \bar{\mathscr{C}}$ de l'arbre étendu, il est donc nécessaire d'évaluer la quantité $S(\bar{w})$ définie par la relation :

$$S(\bar{w}) = \sum_{\tau \in \mathscr{T}_l(\bar{w})} V(\tau)$$

Or, pour chaque sous-arbre τ, la quantité $V(\tau)$ est de la forme :

$$V(\tau) = \prod_{\bar{w}' \in \mathscr{C}(\tau)} V(\bar{w}')$$

et la relation 49 montre que :

$$S(\bar{w}) = \sum_{(\bar{u}_1,\ldots,\bar{u}_p)\, partition\, de\, \mathscr{A}} \prod_{i=1}^{p} S(\bar{u}_i \bar{w})$$

Il suffit par conséquent d'itérer de bas en haut la sommation sur les partitions $\bar{u}_1, \ldots, \bar{u}_p$ possibles sous chaque nœud $\bar{w}$ du produit des sommes $S(\bar{u}_i \bar{w})$ calculées sous chaque nœud.

Cette propriété est quelque peu inattendue, et tient fondamentalement au fait que sommation et produit ne portent pas sur les mêmes entités : il s'agit de sommer sur le *choix de la partition* sous un nœud, alors que le produit porte sur les nœuds impliqués dans une partition particulière. Cependant, disposer de cette propriété permet de transposer l'algorithme CTW au cas des arbres de contexte parcimonieux.

5.2.2. *Algorithme de sommation ascendante.* L'algorithme implémentant le calcul de la constante de normalisation de la distribution a posteriori reprend donc exactement la structure de l'algorithme de maximisation exposé au chapitre précédent. La seule différence est que les maxima sont remplacés par des sommations.

Si en pratique, il reste important de travailler en échelle logarithmique, le choix de cet échelle n'offre pas de simplification des notations comme dans le cas précédent, car nous avons une somme de produits à présent. Nous exposons donc l'algorithme en échelle naturelle.

(1) pour toute feuille $\bar{w} \in \mathscr{C}_l$ de l'arbre étendu, calculer la quantité $V(\bar{w})$

(2) monter d'un niveau, et pour chaque nœud $\bar{w}$ du niveau courant :

   (a) pour chaque partition de l'alphabet $\bar{u}_1, \ldots, \bar{u}_p$ sous le nœud courant, calculer $\prod_{i=1}^{p} S(\bar{u}_i \bar{w})$,

   (b) sommer les quantités obtenues, et stocker cette somme $S(\bar{w})$

(3) retourner en 2 si la racine n'est pas atteinte.

5.3. Intérêt. Nous ne nous étendons pas plus sur cet algorithme CTW parcimonieux, car notre propos est centré sur la sélection de modèle plutôt que leur agrégation. Cependant, disposer de cet algorithme signifie que le calcul de la probabilité d'une séquence sous le modèle bayésien est également possible dans le cadre parcimonieux.

Pour des applications à la compression, cela signifie qu'il est possible de calculer la probabilité de la séquence avec une stratégie efficace, et donc de bénéficier de la quantité accrue d'informations extraite de la séquence par les modèles parcimonieux. Il serait intéressant de mesurer les gains en terme de taux de compression apportés par le recours aux modèles parcimonieux, cependant cela sort du cadre de cette thèse.

Il faut cependant garder à l'esprit que la complexité de l'algorithme de sommation est similaire à celle de l'algorithme de maximisation, et donc très supérieure à celle des algorithmes CTM et CTW : là où ces derniers évaluent un score pour chaque feuille sous chaque nœud, il nous faut ici évaluer un score pour chaque partition possible sous chaque nœud. Il y a donc un coût algorithmique important à l'extension au cadre parcimonieux des méthodes de compression de texte, qui peut être rédhibitoire dans le cas d'une compression en ligne par exemple. Qui plus est, la compression réelle de texte implique de travailler avec un alphabet de 2^8 lettres au minimum (l'ensemble des caractères ASCII, typiquement). Le nombre de partitions à considérer est alors gigantesque, et interdit toute implémentation explicite de ces techniques.

CHAPITRE 9

Evaluation de la sélection de modèle

Nous proposons dans un premier temps un portrait des performances du critère de sélection de modèle proposé, dans diverses situations. Cette évaluation a été menée en deux temps : d'une part, la qualité d'ajustement des modèles parcimonieux sur des séquences biologiques est comparée à celle obtenue par les chaînes de Markov classiques ; d'autre part, une évaluation de la qualité de l'estimation est conduite sur la base de simulations de séquences, sous des modèles appris sur des séquences biologiques.

1. Qualité d'ajustement sur séquences biologiques

1.1. Protocole. Nous montrons dans un premier temps des résultats de comparaison de statistiques de qualité d'ajustement de modèles de dimensions variables. Nous avons retenu le critère BIC comme *arbitre*, compte-tenu de son statut de référence pour la sélection de modèles de complexité bornée.

Nous avons donc procédé à l'évaluation du critère BIC sur l'ensemble des séquences codantes des diverses bactéries répertoriées par le NCBI[**?**]. Nous utiliserons la formulation suivante du critère BIC dans un modèle τ :

$$BIC(\tau) = -\log L(\hat{\boldsymbol{\theta}}_\tau, \boldsymbol{x}) + \frac{|\mathscr{C}(\tau)|(|\mathscr{A}|-1)}{2}\log n$$

où $\hat{\boldsymbol{\theta}}_\tau$ désigne l'estimateur du maximum de vraisemblance dans le modèle τ.

1.2. Résultats. Les graphiques successifs de la figure 4 représentent, pour les différents ordres de 3 à 6, le surcroît de BIC du modèle de Markov par rapport à celui du modèle sélectionné avec une pénalité $k = 0$.

Les points sont étalés en abscisse en fonction du BIC du modèle de Markov. Notons que celui-ci est de l'ordre de la longueur de la séquence, si bien que les points représentant les séquences plus longues sont à droite. Ainsi, le rapport de l'ordonnée par l'abscisse représente la décrue relative du BIC obtenue en substituant le modèle parcimonieux au modèle de Markov classique.

1.3. Commentaires. On remarque en premier lieu que dans l'immense majorité des cas, le modèle parcimonieux sélectionné présente un BIC nettement inférieur à celui obtenu dans le cas d'une modélisation de Markov de même ordre. Cette remarque est même valable systématiquement dans pour les ordres 3 et 6. De plus, l'ordre de grandeur du bénéfice apporté par le recours au modèle parcimonieux est croissant avec l'ordre du modèle, ce qui est satisfaisant.

Par ailleurs, on remarque que l'enveloppe supérieure du nuage de points est bien approchée par une droite de pente négative. Compte-tenu de la relation entre le BIC et la longueur de la séquence, ce signe n'est pas surprenant. En effet, plus la séquence est longue, et plus la chaîne de Markov classique est bien estimée, ce qui explique la décroissance du bénéfice apporté par la sélection d'un modèle parcimonieux.

On remarque par ailleurs que cette pente est d'autant plus faible que l'ordre est élevé. Ce phénomène peut s'expliquer par le précédent : en effet, plus l'ordre est élevé, et plus la longueur de séquence nécessaire à une bonne convergence de l'estimateur de la chaîne de Markov est élevée également.

Ces résultats montrent un intérêt franc pour le recours aux modèles parcimonieux dans le cadre de l'estimation de modèles sur des séquences biologiques. Comme nous

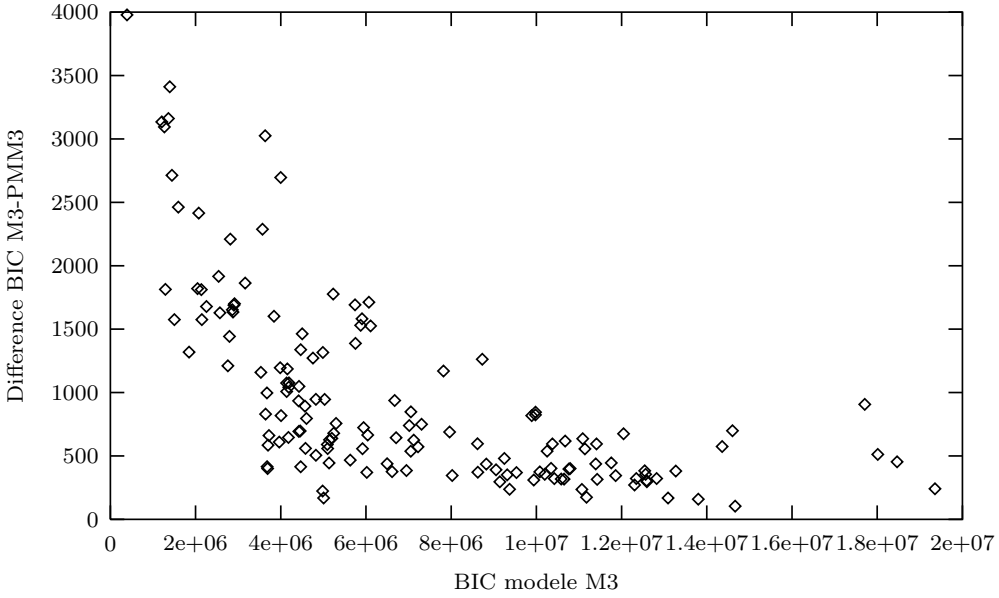


FIG. 1. Différence $BIC(Markov) - BIC(PMM)$ en fonction de $BIC(M)$, calculées sur les séquences codantes des bactéries répertoriées au NCBI. Les modèles estimés sont 3-périodiques afin de mieux représenter la structure des séquences codantes. Les modèles sont d'ordre 3.

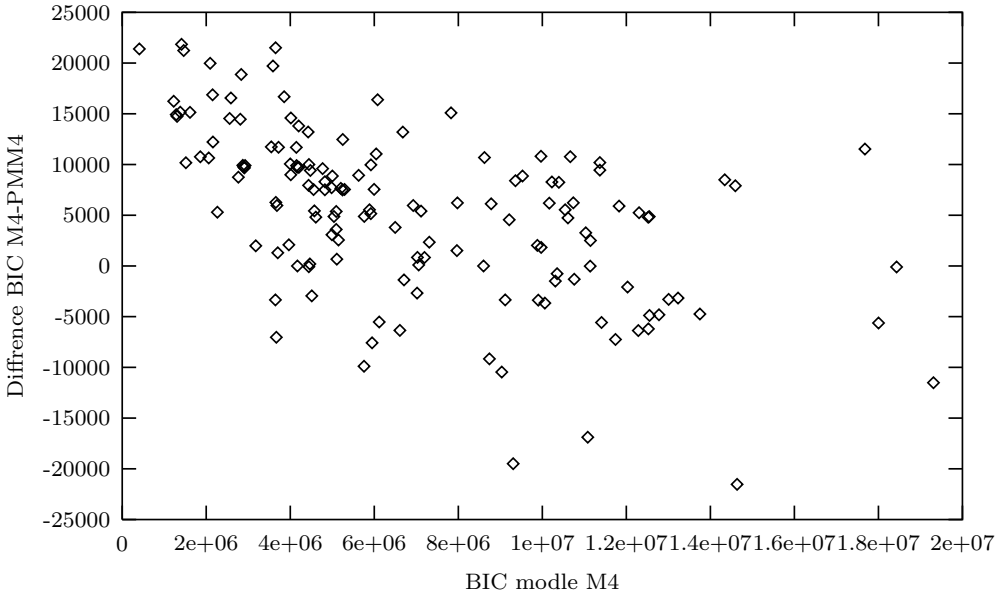


FIG. 2. Différence $BIC(Markov) - BIC(PMM)$ en fonction de $BIC(M)$, calculées sur les séquences codantes des bactéries répertoriées au NCBI. Les modèles estimés sont 3-périodiques afin de mieux représenter la structure des séquences codantes. Les modèles sont d'ordre 4.

le verrons dans la partie suivante, ce bénéfice n'est pas seulement dû à la pénalisation de la chaîne de Markov.

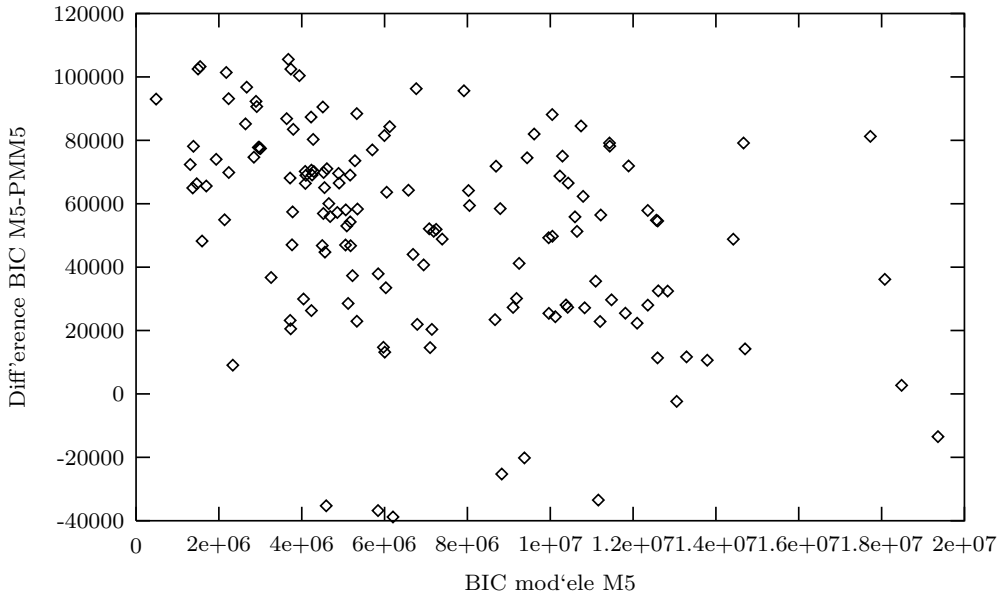


FIG. 3. Différence $BIC(Markov) - BIC(PMM)$ en fonction de $BIC(M)$, calculées sur les séquences codantes des bactéries répertoriées au NCBI. Les modèles estimés sont 3-périodiques afin de mieux représenter la structure des séquences codantes. Les modèles sont d'ordre 5.

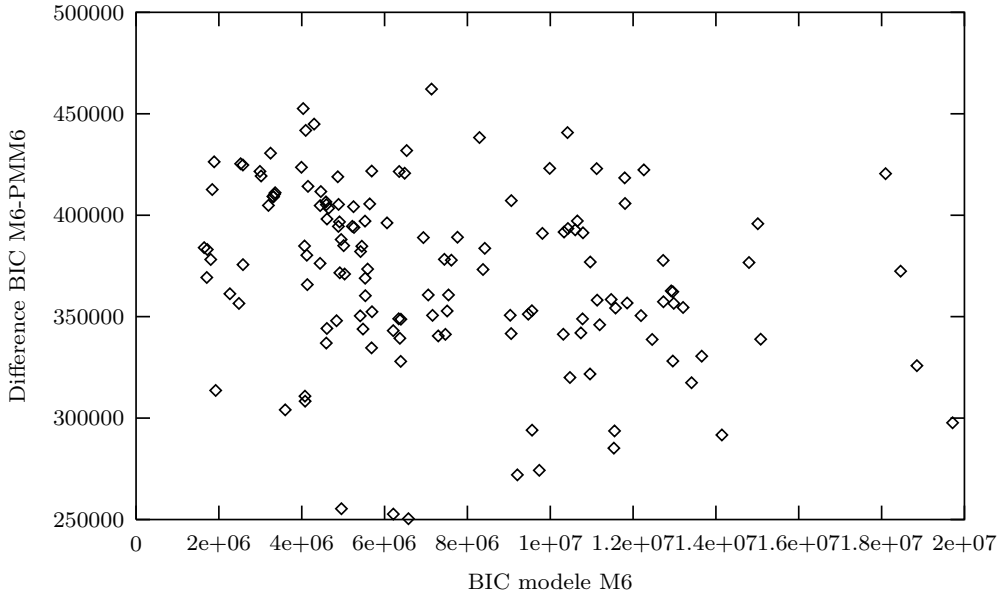


FIG. 4. Différence $BIC(Markov) - BIC(PMM)$ en fonction de $BIC(M)$, calculées sur les séquences codantes des bactéries répertoriées au NCBI. Les modèles estimés sont 3-périodiques afin de mieux représenter la structure des séquences codantes. Les modèles sont d'ordre 6.

2. Qualité d'estimation

Nous nous intéressons à présent à un autre aspect de qualité d'un critère de sélection de sous-modèle, à savoir la qualité d'estimation qu'il permet. Le leitmotiv de ce chapitre, au-delà du souci de comparaison aux modèles classiques, est de fournir à la communauté bioinformaticienne un guide pour le choix de l'ordre des chaînes de Markov employées. Le discours adressé est le suivant : si les distribution estimées en pratique étaient effectivement les distributions sous-jacentes aux séquences biologiques, alors la variabilité de l'estimateur donne toutes les chances d'estimer une matrice de transition erronnée.

2.1. Protocole. En pratique, nous avons retenu le protocole suivant. Sur les séquences codantes de la bactérie *Escherichia Coli*, nous avons estimé la matrice de Markov de la phase 1 (rappelons ici que les séquences codantes sont modélisées par des modèles 3-périodiques, où une matrice de transition différente est utilisée pour générer les lettres selon le résidu modulo 3 de leur position). Nous avons estimé des modèles d'ordre élevé (5), conformément aux pratiques courantes en bioinformatique. L'expérience a également été conduite en utilisant un modèle de Markov parcimonieux comme référence, afin de mesurer l'impact de la présence de contraintes parcimonieuses sur les résultats.

Ces modèles ont ensuite été utilisés pour générer des séquences aléatoirement, pour diverses longueurs (10^4 et 10^5 caractères). Nous avons estimé un modèle de Markov classique d'une part, et un modèle parcimonieux d'autre part, sur chacune des séquences simulées. En fait, plusieurs modèles parcimonieux de chaque ordre ont été estimés, selon diverses valeurs de la pénalité k.

Nous avons alors évalué, pour chacune des simulations et chacun des modèles estimés, la distance en variation totale entre la distribution estimée et la distribution *source*.

2.2. Résultats. Nous avons représenté sur les graphiques les résultats obtenus pour les diverses longueurs de séquences. Sur chaque graphique, l'abscisse représente l'ordre du modèle estimé, et l'ordonnée la distance en variation totale moyenne entre la distribution estimée et la source. La moyenne est donc prise sur l'ensemble des simulations.

Ainsi, la figure 2 présente les distances moyennes obtenues pour des séquences de longueur 10^4, simulées sous un modèle de Markov parcimonieux d'ordre 5. La figure 3 présente les mêmes résultats, mais dans le cas de séquences de longueur 10^5. Les figures 4 et 5, enfin, présentent les mêmes résultats dans le cas d'un modèle de référence qui n'est pas parcimonieux.

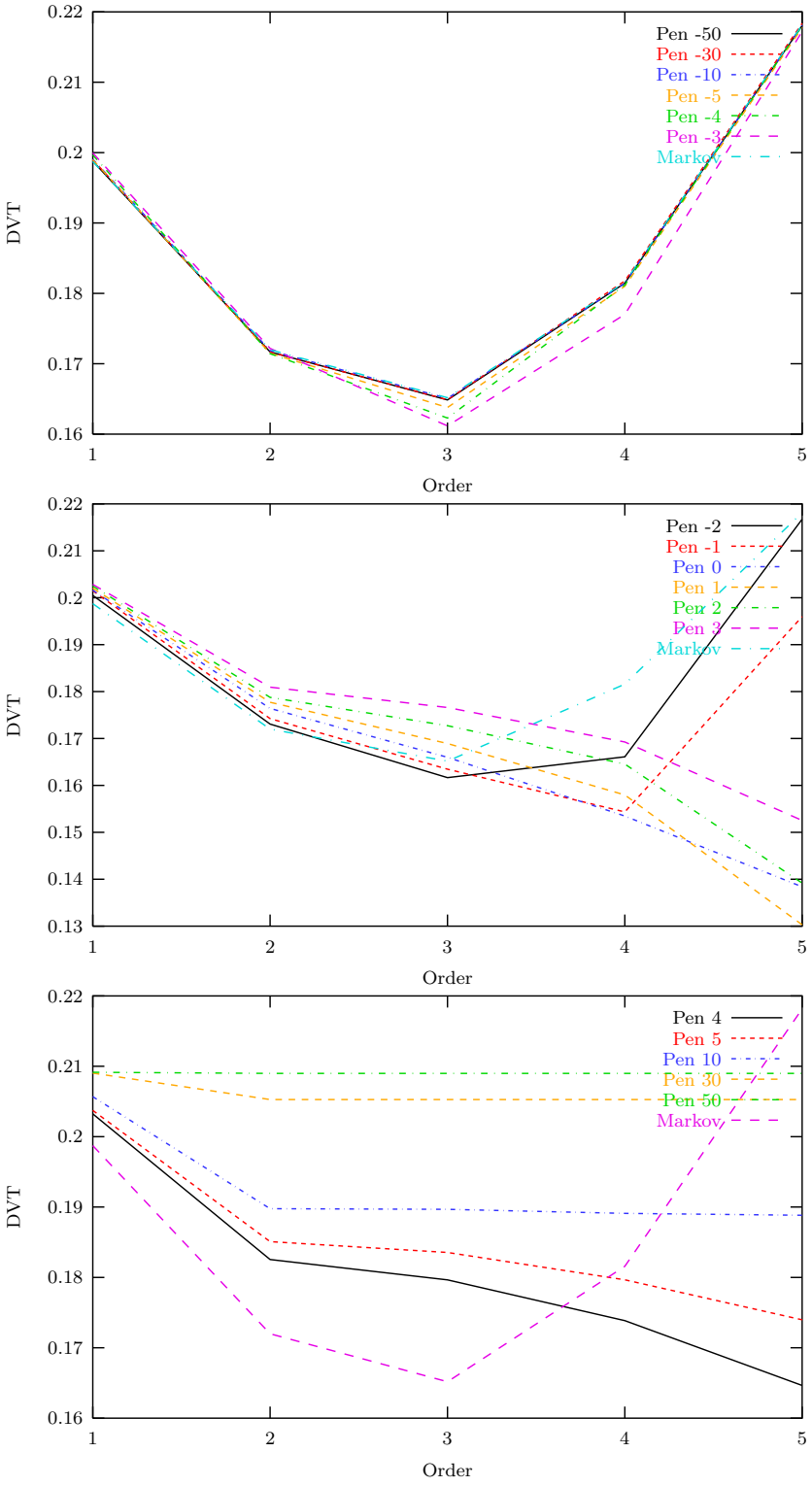


FIG. 5. Distance en variation totale Estimateur/Source, en moyenne sur 100 simulations. Le modèle simulé est parcimonieux d'ordre 5, estimé sur des données biologiques. Les séquences simulées ont une longueur de 10^4 lettres.

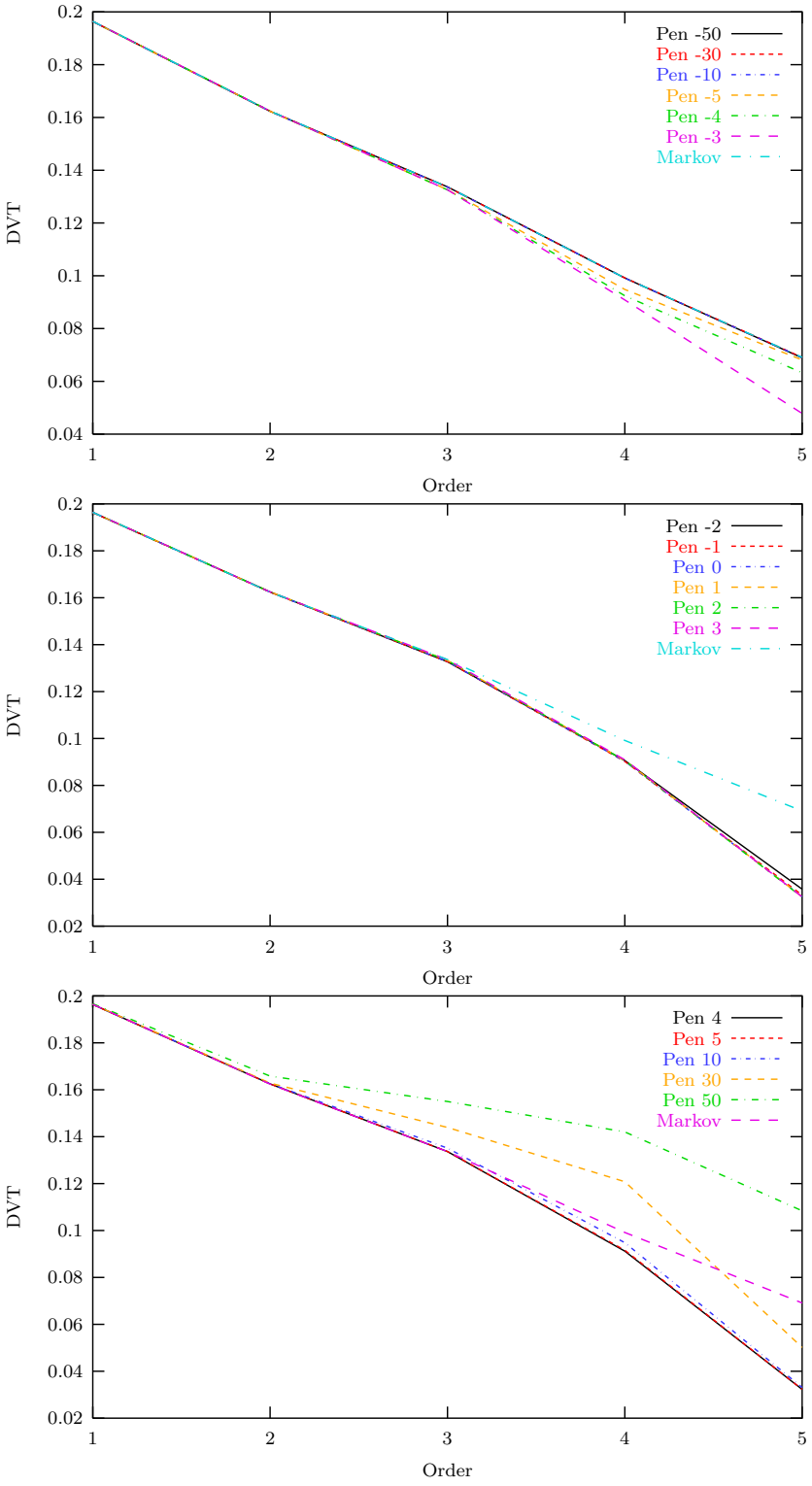


FIG. 6. Distance en variation totale Estimateur/Source, en moyenne sur 100 simulations. Le modèle simulé est parcimonieux d'ordre 5, estimé sur des données biologiques. Les séquences simulées ont une longueur de 10^5 lettres.

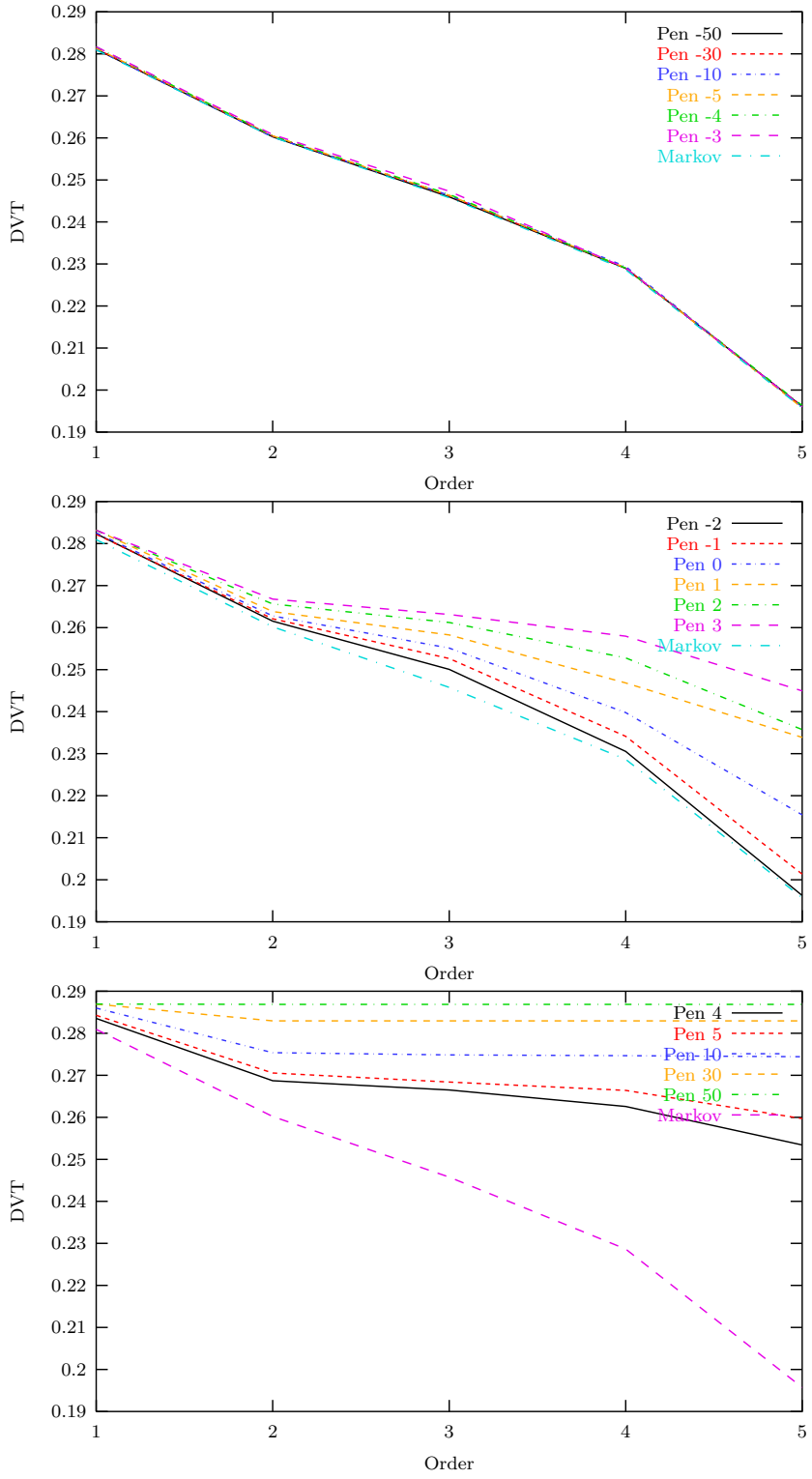


FIG. 7. Distance en variation totale Estimateur/Source, en moyenne sur 100 simulations. Le modèle simulé est markovien d'ordre 5, estimé sur des données biologiques. Les séquences simulées ont une longueur de 10^4 lettres.

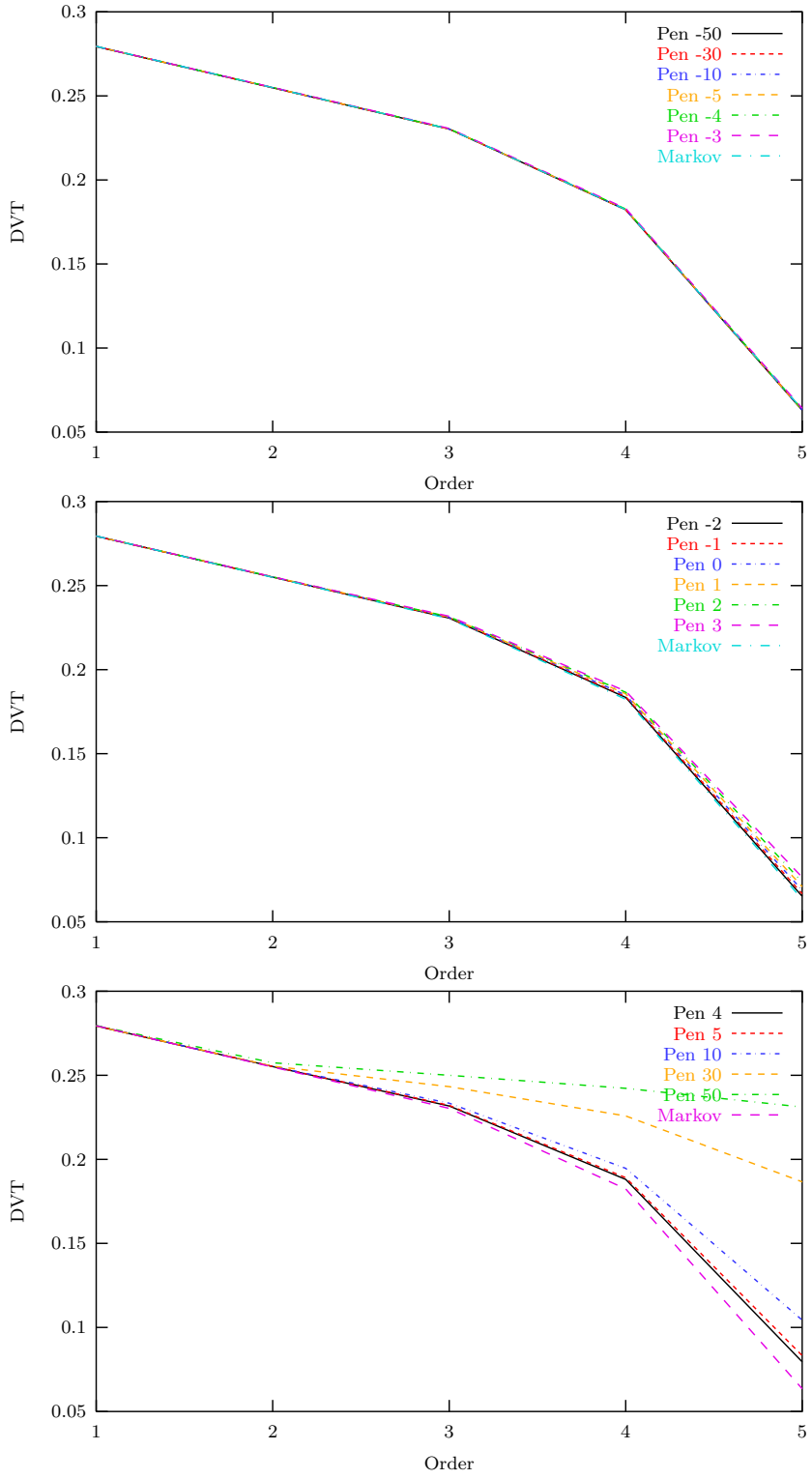


FIG. 8. Distance en variation totale Estimateur/Source, en moyenne sur 100 simulations. Le modèle simulé est markovien d'ordre 5, estimé sur des données biologiques. Les séquences simulées ont une longueur de 10^5 lettres.

2.3. Commentaires. L'expérience menée comporte deux dimensions : la longueur de la séquence d'une part, et la complexité du signal simulé d'autre part. Nous analysons l'influence de ces deux dimensions sur les performances de la sélection de modèle bayésienne séparément. Cependant, on constate rapidement plusieurs comportements de la qualité d'estimation :

- sur tous les graphiques, des pénalités raisonnables (de l'ordre de quelques unités) permettent d'obtenir une courbe décroissante en fonction de l'ordre, ce qui est très satisfaisant
- la perte de qualité d'estimation par rapport à la chaîne de Markov d'ordre optimal est limitée,
- la pénalité nulle $k = 0$ fournit de bons résultats, ce qui limite les problèmes liés à la sélection de la pénalité.

2.3.1. *Influence de la pénalité.* Dans l'ensemble des cas simulés, nous avons réalisé la sélection de modèles pour diverses valeurs de la pénalité $\log k$. Le comportement du faisceau de courbes obtenu est conforme aux attentes :

- pour des pénalités très faibles ($\log k = -50$), on retrouve le comportement de la chaîne de Markov classique de même ordre. Ce comportement se retrouve d'ailleurs sur des séquences plus courtes, et montre que des pénalités négatives de l'ordre de quelques dizaines suffisent à forcer la sélection des arbres les plus grands sur des séquences d'une longueur de l'ordre de 10 000.
- pour des pénalités proches de 0, on obtient des courbes qui réalisent un compromis intéressant entre complexité du modèle et qualité d'estimation,
- pour des pénalités très élevées ($\log k = 50$), on obtient une courbe quasiment constante, traduisant un élagage drastique de l'arbre de contexte parcimonieux.

Cependant, il apparaît que la pénalité nulle ne présente pas toujours un comportement totalement satisfaisant lorsque l'ordre croît. En effet, sur la figure 5, on remarque que la pénalité $\log k = 1$ réalise un meilleur compromis à l'ordre 5 que la pénalité nulle. La pénalité nulle reste cependant, dans l'écrasante majorité des cas, le choix globalement optimal par rapport aux variations de l'ordre.

2.3.2. *Influence de la longueur de la séquence.* Une caractéristique des approches bayésiennes est leur équivalence asymptotique avec les méthodes classiques. Face à la croissance de la taille de l'échantillon observé, le choix de la distribution a priori influencent de moins en moins la distribution a posteriori.

Ce phénomène se retrouve bien dans les données simulées, à travers une plus grande dispersion des courbes associées aux diverses pénalités sur des séquences de longueur 10^4 que sur des séquences de longueur 10^5.

Par ailleurs, on remarque que l'écart entre la courbe de l'estimateur de Markov classique et celui de l'estimateur PMM avec pénalité nulle se réduit avec l'augmentation de la taille de la séquence. Ce phénomène, déjà observé au chapitre précédent, est lié à la croissance de la qualité d'estimation du modèle de Markov classique avec la taille de la séquence, qui réduit par conséquent le bénéfice apporté par les modèles parcimonieux.

Enfin, on constate une différence essentielle entre les courbes obtenues pour les longueurs 10^4 et 10^5 : dans le premier cas, la courbe de l'estimateur de la chaîne de Markov classique, ainsi que celles des estimateurs PMM avec pénalité négative, deviennent croissante à partir de l'ordre 3, contrairement à celles obtenues pour les pénalités nulles ou positives. Cette observation montre que la sélection de modèle permet effectivement de n'ajouter que les paramètres nécessaires, comme l'illustre la décroissance de la courbe en fonction de l'ordre pour la pénalité nulle observée sur les figures 5 et 7.

2.3.3. *Influence de la complexité du signal.* Nous discutons enfin l'influence de la complexité de la source du signal, que nous avons fait varier en simulant sous une chaîne de Markov d'ordre 5 d'une part (figures 7 et 8), et sous un modèle parcimonieux de même ordre d'autre part (figures 5 et 6).

Les comportements obtenus pour la courbe représentant la qualité d'estimation de l'estimateur classique de la chaîne de Markov dans ces deux situations sont assez différents, du moins pour des séquences simulées de longueur 10^4 (figures 5 et 7). En effet, cet estimateur montre une dégénérescence de la qualité d'estimation lorsque l'ordre dépasse 3 dans les simulations sous le modèle parcimonieux, mais pas dans celles sous le modèle classique.

Cela montre que le recours aux modèles parcimonieux n'est pas nécessairement bénéfique dans le cas d'un signal d'une complexité élevée, même si l'estimateur obtenu dans le modèle parcimonieux sélectionné atteint des performances comparables.

2.4. Conclusion. Il apparaît donc que la sélection d'un modèle parcimonieux permet d'améliorer la qualité d'estimation dans de nombreuses situations, et principalement face à des séquences courtes. De plus, dans les cas où ils ne présentent pas des performances supérieures à l'estimateur classique des chaînes de Markov, ils atteignent quasiment systématiquement des performances voisines.

De plus, il apparaît que les meilleurs compromis sont obtenus pour des pénalités d'amplitude faible (de l'ordre de quelques unités), et que la pénalité nulle assure souvent un très bon compromis. Ce point est rassurant, dans la mesure où il justifie de ne pas réaliser une sélection de la pénalité dans les cas pratiques. Nous sommes en effet actuellement démunis en terme de critère de choix de cette pénalité a priori sur les modèles.

Il manque dans ce comparatif les chaînes de Markov à longueur variable, pour lesquelles il était nécessaire de réaliser une implémentation du choix du seuil dans la règle d'élagage. Nous prévoyons cependant de réaliser une évaluation plus exhaustive, afin de délimiter nettement les conditions dans lesquelles le recours à ces procédures de sélection de modèle se justifie.

Enfin, le comportement peu satisfaisant de l'estimateur dans le modèle sélectionné obtenu dans le cas d'un signal d'ordre élevé interroge sur la pertinence d'une sélection de modèle seule. On pourrait en effet envisager d'autres manières de construire un estimateur des transitions dans la séquence, par exemple en considérant la moyenne du paramètre contre la distribution a posteriori.

3. Application à la classification des protéines

Un cadre où l'on s'attend à bénéficier pleinement de la parcimonie est l'analyse des séquences de protéines. Les protéines sont en effet constituées d'acides aminés, qui sont en nombre bien plus grands que les nucléotides. Une protéine est également trois fois plus courte que le gène qui la code. Aussi, on s'attend à tirer meilleur avantage de la parcimonie pour l'analyse des protéines que pour la détection des gènes.

L'écueil évident de l'application des modèles parcimonieux à l'analyse des protéines est le temps de calcul et l'espace mémoire requis. Cet écueil n'est cependant pas rédhibitoire, et Florencia LEONARDI a pu l'appliquer avec succès à la classification de familles de protéines en recourant à une variante de l'algorithme présenté précédemment. Ces travaux ont été publiés dans la revue *Bioinformatics* (voir [**?**]).

Une piste alternative envisagée un temps pour contourner l'écueil combinatoire est de regrouper a priori les acides aminés pour former les contextes. En d'autres termes, il s'agit de restreindre l'ensemble des partitions de l'alphabet envisagées. Cela peut être réalisé en s'inspirant des classifications physico-chimiques des acides aminés.

Les différents acides aminés présentent en effet des propriétés physico-chimiques variées, mais selon lesquelles il est possible de les regrouper en fonction de leur ressemblance. Il existe plusieurs telles classifications : en fonction de l'hydrophobicité, de la taille, de la polarité, ou encore des groupes fonctionnels ou entités chimiques qu'ils contiennent.

Il est cependant possible de réaliser une classification des différents acides aminés en 9 classes, à partir desquelles il est possible de reconstituer l'ensemble des classes proposées par les biochimistes par union de ces 9 classes. Une telle classification conduit à distinguer :

- Alanine, Leucine, Isoleucine, Valine
- Arginine et Lysine
- Asparagine et Glycine
- Aspartate et Glutamate
- Cystéine et Méthionine
- Glutamine et Thréonine
- Histidine
- Phenylalanine, Tyrosine et Tryptophane
- Proline
- Serine

Pour utiliser une telle classification dans le cadre des modèles parcimonieux, il suffit de n'envisager que les partitions de l'alphabet qui ne séparent jamais deux acides aminés appartenant à la même classe. Ce choix est donc arbitraire, mais motivé par la connaissance des propriétés chimiques des molécules.

L'opportunité de mettre en œuvre une telle approche ne s'est pas présentée, mais la possibilité existe dans l'implémentation de l'algorithme de sélection de modèles incluse dans la librairie logicielle `seq++` (`http ://stat.genopole.cnrs.fr/seqpp`, voir aussi l'*application note* dans *Bioinformatics*, [**?**]).

Conclusion et perspectives

Deux regrets...

Le maximum d'entropie comme unification. La concentration de l'entropie est un résultat de nature combinatoire, qui exprime que parmi toutes les configurations possibles d'un système, celle dont les fréquences des états coïncide avec les fréquences prédites par la distribution de maximum d'entropie peut être réalisée d'un plus grand nombre de manières que toutes les autres. Ce résultat complète la justification du recours au principe de maximum d'entropie, et constitue même l'argument principal de la physique statistique de BOLTZMANN (cf. la fameuse formule $S = k_B \ln W$, qui exprime que l'entropie d'un système est proportionnelle au logarithme du nombre d'états du système conduisant à la même configuration, c'est-à-dire indistinguables les uns des autres).

Ce résultat sous tend l'ensemble des résultats de convergence en statistique, puisqu'il permet de majorer la probabilité qu'un estimateur dans un modèle de maximum d'entropie ne converge pas vers sa *vraie* valeur. Il serait satisfaisant de disposer d'une dérivation de la consistance de l'estimateur du maximum de vraisemblance dans les modèles markoviens à partir de ce seul principe. Cependant, la signification profonde de ce résultat de concentration m'échappe encore, et ce terrain m'est apparu, à l'heure actuelle, trop glissant pour m'y aventurer. Cette ambition reste cependant d'autant plus présente que la concentration de l'entropie m'apparaît comme une propriété tout à fait fondamentale de la théorie *jaynésienne* des statistiques.

Un autre aspect qui aurait contribué à unifier la théorie des chaînes de MARKOV avec le principe de maximum d'entropie aurait été de calculer les estimateurs des paramètres lagrangiens de la distribution de maximum d'entropie par la relation usuelle de la physique statistique, à savoir l'adéquation entre les comptages observés et la dérivée de la fonction de partition :

$$\forall j \in \{1,\dots,k\}, \forall \boldsymbol{a} \in \mathscr{A}^j, \frac{\partial \ln Z}{\partial \mu_{\boldsymbol{a}}} = N_{\boldsymbol{a}}(\boldsymbol{x})$$

où $\boldsymbol{x} \in \mathscr{A}^n$ désigne l'échantillon sur lequel l'estimation a lieu.

La géométrie des paramètres d'un modèle de maximum d'entropie. Comme nous l'avons vu dans la première partie de ce manuscrit, la maximisation de l'entropie passe inévitablement par la maximisation d'un Lagrangien. Ce faisant, on introduit des multiplicateurs de *Lagrange* comme variables additionnelles, et ces multiplicateurs coïncident précisément avec les paramètres des modèles statistiques résultant de la maximisation de l'entropie (autrement dit, ils ne sont rien d'autre que le paramètre consacré θ de la statistique classique).

Cependant, dans le cas des chaînes de MARKOV, il apparaît que la prise en compte des comptages des mots de 1 **et** 2 lettres conduit à un paramètre plus richement structuré que les seules entrées de la matrice de transition. En effet, si l'on note $\pi(i,j)$ la probabilité $\mathbb{P}(X_t = j | X_{t-1} = i)$, et $\mu_i, \mu_j, \mu_{i,j}$ les multiplicateurs de LAGRANGE associés aux comptages de i, j et ij respectivement, on a vu par identification des modèles que :

$$\pi(i,j) = \exp -\mu_j - \mu_{i,j}$$

Autrement dit, le paramétrage résultant de la maximisation de l'entropie reconnaît deux contributions dans la probabilité de transition : l'une provenant du fait que la lettre j apparaît, et l'autre prenant en compte le fait qu'elle apparaît *après* j. En fait, le terme $\mu_{i,j}$ apparaît comme le terme correctif à apporter à la distribution d'indépendance (autrement dit la distribution de maximum d'entropie pour l'observation de la composition en lettres de la séquence) pour prendre en compte les comptages des mots de deux lettres. Ceci apparaît clairement lorsque l'on s'intéresse à la distribution de probabilité sur une séquence de seulement 2 lettres :

$$\forall i, j \in \mathscr{A}^2, \mathbb{P}(X_1 = i, X_2 = j) = \exp -\mu_i - \mu_j \times \exp -\mu_{i,j}$$

Dans le premier terme du second membre, on reconnaît clairement une distribution d'indépendance, alors que le second terme vient *corriger* cette distribution pour refléter les dépendances d'ordre 1 dans la séquence telles que rapportées par les comptages des mots de deux lettres.

L'obstacle principal à cette interprétation des paramètres issus de la maximisation de l'entropie sous la contrainte des comptages des mots de 1 et 2 lettres est d'établir que les multiplicateurs de *Lagrange* associés aux comptages des lettres seules prennent la même valeur, que les comptages de mots de 2 lettres soient pris en compte ou non. Je n'y suis pas parvenu, même si je reste convaincu que cette propriété est vérifiée.

Cela aurait deux conséquences principales : la première renvoie à l'approche classique, dans laquelle un débat persiste sur l'estimation de la distribution initiale. L'approche probabiliste des chaînes de MARKOV rend en effet toute la distribution sur les séquences dépendante du choix de cette distribution initiale, même si le caractère mélangeant des chaînes de *Markov efface* cette dépendance à force d'itérer les transitions. Si la propriété suggérée précédemment est vraie, alors le principe de maximum d'entropie permet d'établir que la *bonne* distribution initiale à considérer est la distribution stationnaire associée aux probabilités de transition estimées. Si ce choix semble *raisonnable*, je n'ai pas connaissance d'un argument décisif en sa faveur, et il serait très satisfaisant d'en dériver un du principe de maximum d'entropie.

La seconde conséquence concerne la décomposition des dépendances dans une chaîne de MARKOV. En effet, nous avons vu ci-dessus que le multiplicateur de LAGRANGE $\mu_{i,j}$ associé aux comptages des mots de 2 lettres apportait la *correction* nécessaire au modèle d'indépendance permettant de refléter les comptages des mots de 2 lettres. Cela peut d'ailleurs se généraliser à des dépendances d'ordre supérieur, car :

$$\forall (i_1,\dots,i_k)\in\mathscr{A}^k, \mathbb{P}(X_1=i_1,\dots,X_k=i_k)=\exp-\sum_{a\in\mathscr{A}}\mu_a N_a(\boldsymbol{X})-\dots-\sum_{\boldsymbol{a}\in\mathscr{A}^k}\mu_{\boldsymbol{a}}N_{\boldsymbol{a}}(\boldsymbol{X})$$

Autrement, les termes $\mu_{a_1,\dots,a_j}$, $j\leq k$, représentent le correctif nécessaire au modèle d'ordre $j-1$ pour qu'il reflète les comptages des mots de longueur j.

Il se trouve qu'AMARI, l'une des références en géométrie de l'information, avait recherché de telles décompositions des paramètres d'une chaîne de MARKOV dans l'article intitulé *Information geometry on hierarchical decomposition of stochastic interactions* [**?**]. Cependant, son point de départ était le paramétrage par les probabilités de transition d'une chaîne de MARKOV plutôt que les multiplicateurs de LAGRANGE issus de la maximisation de l'entropie, mais son objectif est de paramétrer les dépendances itérativement, les premiers paramètres désignant le modèle d'indépendance le plus proche, puis les suivants venant *corriger* ce modèle pour prendre en compte les dépendances d'ordres de plus en plus grands. Il caractérise ces paramètres d'ordre supérieur comme représentant *purement* les dépendances à cet ordre, celles d'ordre inférieur étant déjà modélisées par les paramètres précédents. Formulé par des projections, canoniques pour la géométrie de l'information, sur les modèles d'ordre inférieur, il résoud en quelque sorte le problème inverse de celui présenté ici en identifiant les paramètres issus du maximum d'entropie avec les probabilités de transition.

Ces deux approches doivent clairement être réconciliées, et je souhaite prochainement apporter une réponse à la question de savoir si les paramètres issus du maximum d'entropie coïncident avec ceux proposés par AMARI. Il est à ce titre intéressant de remarquer qu'incidemment, dans cet article, AMARI place la remarque que tous les modèles qu'il envisage maximisent l'entropie parmi l'ensemble des modèles sur l'espace d'état considéré. Mais cette remarque occupe deux lignes. Il semble qu'il ait intuité la même remarque que celle présentée ici, car rien ne justifie vraiment cette précision dans le cours de son exposé.

... et une satisfaction

Cette initiation personnelle au principe de maximum d'entropie m'a permis de trouver les fondements de la démarche statistique recherchés depuis le début de ma formation de statisticien. Mais au-delà de la satisfaction intellectuelle apportée par cette découverte pour moi, la connaissance de cette théorie du maximum d'entropie apporte une confiance accrue dans le choix des modèles face à des espaces d'états non conventionnels.

Un exemple concret de cela m'a été apporté par mon travail au Génoscope sur les modèles du métabolisme, dans lesquels un état métabolique d'une cellule est représenté par les flux des réactions chimiques qui s'y déroulent. L'espace d'état est par ailleurs contraint de respecter les lois fondamentales de la physique, telles que la conservation de la masse ou la seconde loi de la thermodynamique (dès lors que des concentrations intra-cellulaires de composés chimiques sont connues, et incluses dans l'espace d'états).

Jusqu'à présent, l'essentiel de la littérature dans ce domaine a totalement négligé le fait que les mesures expérimentales sur lesquelles sont fondées ces modèles sont réalisées sur des colonies de cellule, et non sur des cellules uniques. Or, les contraintes qui façonnent l'espace d'état s'entendent fondamentalement, elles, comme s'appliquant à l'échelle des individus. Il se trouve que tant qu'aucune concentration intracellulaire n'est mesurée, l'espace d'état est convexe, et donc il est possible de considérer un *individu moyen* représentatif de la colonie sans perte de généralité. Mais dès lors qu'il s'agit de comparer les états métaboliques entre deux colonies, il est bien évidemment délicat de déterminer la significativité des écarts entre ces individus moyens sans probabiliser l'espace d'état.

Compte-tenu de la structure complexe de cet espace (rappelons qu'il résulte d'un ensemble de contraintes traduisant les lois physiques), aucune distribution de probabilité ne s'impose immédiatement : les distributions usuelles correspondent en effet à des espaces d'états simples et récurrents, typiquement $\mathbb{N}, \mathbb{R}, \mathbb{R}_+, \ldots$. En revanche, le principe de maximum d'entropie peut aussi bien s'appliquer dans ce cas que dans d'autres : on a un espace d'états d'une part, des observations moyennes (ou cumulées) sur la population d'autre part. On peut donc caractériser la distribution de maximum d'entropie sur l'espace d'états sous la contrainte qu'elle prédise les valeurs moyennes observées expérimentalement.

Mais deux difficultés se présentent : l'espace d'états est un sous-espace d'un espace continu, et il faut donc choisir une mesure de référence contre laquelle maximiser l'entropie. En l'occurrence, le choix de la mesure de LEBESGUE restreinte à l'espace d'états semble raisonnable. L'autre difficulté est d'ordre analytique : il est bien entendu difficile d'évaluer la fonction de partition, puisqu'il faut intégrer la densité de probabilité $\exp -\sum_{r\in R} \mu_r \nu_r$, où R désigne l'ensemble des réactions chimiques ayant lieu dans l'organisme, sur un sous-espace défini par environ un millier de contraintes linéaires. En l'absence de solution analytique à ce problème, le seul salut réside dans le recours aux méthodes simulées. On dispose en effet de la distribution de maximum d'entropie à une constante près (puisque seule la fonction de partition requiert une intégration), et les méthodes MCMC permettent de simuler de telles distributions. La difficulté réside alors dans la détermination des valeurs des paramètres.

Annexes : autres travaux

ANNEXE A

An EM algorithm for estimation in the Mixture Transition Distribution model, *Journal of Statistical Computations and Simulations*

Mentionnées à plusieurs reprises au long de ce manuscrit, les modèles MTD constituent une manière de réduire la dimension des modèles de MARKOV *d'ordre élevé différente des arbres de contexte. Alors que les arbres de contexte n'utilisent comme observables que des* mots*, au sens de sous-séquences formées de symboles successifs, les MTD s'attachent plutôt à modéliser les interactions dans la séquence comme une superposition d'interactions deux-à-deux entre des symboles éloignés d'un petit nombre de positions dans la séquence. Cette superposition est effectuée de manière additive.*

L'article qui suit résulte d'une étude conjointe avec Sophie LÈBRE*, qui constituait le sujet de son stage de DEA au laboratoire Statistique et Génome en 2004. Cette étude s'était focalisée principalement sur deux aspects relativement disjoints de ce modèle : l'absence d'un paramétrage identifiable d'une part, et l'absence de méthode* standard *pour mener son estimation.*

Le premier de ces points est intimement lié à la formulation originale du modèle, lequel y est paramétré de manière redondante : diverses valeurs du paramètre définissent, en réalité, la même distribution de probabilité. Bien que l'absence d'un paramétrage identifiable ne soit pas rédhibitoire pour l'exploitation utile d'un modèle statistique, il reste qu'elle complique la comparaison des paramètres estimés, et que, plus généralement, elle compromet la possibilité de recourir à l'arsenal usuel de résultats asymptotiques valables pour les modèles identifiables réguliers. Il est rapidement apparu qu'il n'existe pas de solution triviale à ce problème, même s'il s'est révélé possible de substituer au paramétrage proposé initialement un nouveau paramètre de dimension inférieure. Cependant, ce paramètre n'est également pas identifiable.

Le second point, l'estimation dans le modèle MTD, a suscité l'intérêt suite à deux réflexions : tout d'abord, le problème d'optimisation de la vraisemblance de ce modèle n'admet pas de solution analytique explicite, si bien que les méthodes proposées dans la littérature pour le résoudre s'appuient sur des algorithmes numériques d'optimisation sous contraintes ; d'autre part, une interprétation du modèle MTD comme un modèle à variable caché (la variable cachée étant celle indiquant, en chaque position, la distance à laquelle se trouve le symbole en interaction avec le symbole courant). Il nous est alors apparu naturel de transposer l'algorithme EM, présenté dans la première partie de ce manuscrit, et très populaire pour les problèmes d'estimation de modèles à variables cachées, à ce modèle.

Ces deux contributions sur les modèles MTD sont en cours de publication dans la revue Journal of Statistical Computations and Simulations*, dans un article proposé ci-après.*

Considérée aujourd'hui sous le jour du principe de maximum d'entropie, les modèles MTD apparaissent étranges. On s'attend en effet, compte-tenu de la formulation du modèle, à ce que l'on puisse l'estimer en quelque sorte comme un modèle de Markov usuel, c'est-à-dire avec les fréquences d'occurrence $(n_{ab}^k)_{(a,b)\in\mathscr{A}^2,0<k\leq l}$ *des couples de lettres* $(a,b)\in\mathscr{A}^2$ *séparés de* k *positions,* k *variant de 0 à l'ordre maximal du modèle* l*.*

Si l'on s'en tient à ces quantités, comme un choix d'observables, alors on peut chercher à appliquer le principe de maximum d'entropie. Qui conduira inévitablement à considérer la distribution suivante :

$$\forall \boldsymbol{x} \in \mathscr{A}^n, \mathbb{P}(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{\mu})} \exp - \sum_{(a,b)\in\mathscr{A}^2} \sum_{k=1}^{l} \mu_{ab}^k n_{a,b}^k(\boldsymbol{x})$$

avec $\boldsymbol{\mu} = (\mu_{a,b}^k)_{(a,b)\in\mathscr{A}^2, 0<k\leq l}$ *les multiplicateurs de* LAGRANGE *associés à ces comptages. Aucune investigation d'un tel modèle ne nous est connue, et en mener une compte parmi les projets de l'auteur.*

An EM algorithm for estimation in the Mixture Transition Distribution model

Sophie Lèbre**, Pierre-Yves Bourguignon
Laboratoire Statistique et Génome,
UMR 8071 Université Evry Val d'Essonne/CNRS/INRA
523, place des Terrasses de l'Agora, 91000 Evry, France.

October 20, 2006

Abstract

The Mixture Transition Distribution (MTD) model was introduced by Raftery to face the need for parsimony in the modeling of high-order Markov chains in discrete time. The particularity of this model comes from the fact that the effect of each lag upon the present is considered separately and additively, so that the number of parameters required is drastically reduced. Nevertheless, the estimation of the MTD still remains problematic on account of the large number of constraints on the parameters. To face such difficulties, we propose to come down to a better known problem: estimation of incomplete data by an Expectation-Maximization (EM) algorithm. In this paper, we develop an iterative procedure to estimate MTD parameters offering the convergence properties of an EM algorithm. Estimations of high-order MTD models led on DNA sequences outperform the corresponding fully parametrized Markov chain in terms of Bayesian Information Criterion.

A software implementation of our algorithm is available in the library seq++ at `http://stat.genopole.cnrs.fr/seqpp`.

keywords: Markov chain; mixture transition distribution (MTD); Parsimony; Maximum likelihood; EM algorithm;

1 Introduction

While providing an efficient framework for discrete-time sequence modeling, higher-order Markov chains suffer from the exponential growth of the parameter space dimension with respect to the order of the model, which results in the inaccuracy of the parameters estimation when a limited amount of data is available. This fact motivates the development of approximate versions of higher-order Markov chains, such as the Mixture Transition Distribution (MTD) model [1, 2]. Thanks to a simple structure, where each lag contributes to the prediction of the current letter in a separate and additive way, the dimension of this model's parameter space grows only linearly with respect to the order. Variable length Markov chains [**?**] are another example of such an approximation.

** To whom correspondence should be addressed.

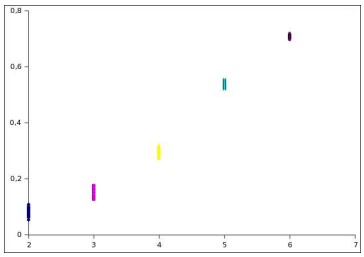


Figure 1: Total variation distance between distributions estimated from randomly generated sequences and the generating distribution. The generating model is of order 5, and the random sequences are 5000 letters long. The abscissa represents the order of the estimated Markov chain.

Nevertheless, maximum likelihood estimation in the MTD model is subject to such constraints that analytical solutions are beyond the reach of present methods. One has thus to retort to numerical optimization procedures. The main method proposed to this day is due to Berchtold [3], and relies on an ad-hoc optimization method. The main contribution of this paper is to fit the MTD model into the general framework of hidden variable models, and derive a version of the classical EM algorithm for the estimation of its parameters.

In this first section, we define the MTD model and recall its main features and some of its variants. Parameterization of the model is discussed in section 2, where we establish that under the most general definition, it is not identifiable. Then we shed light on an identifiable set of parameters. Derivations of the update formulas involved by the EM algorithm are detailed in section 3. We finally illustrate our method by some applications to biological sequence modeling.

Need for parsimony Markov models are pertinent to analyze m-letters words composition of a sequence of random variables [**?**, **?**]. Nevertheless, the length m of the words the model accounts for has to be chosen by the statistician. On the one hand, a high order is always preferred since it can capture strictly more information. On the other hand, since the parameter's dimension increases exponentially fast with respect to the order of the model, higher order models cannot be accurately estimated. Thus, a trade-off has to be drawn to optimize the amount of information extracted from the data.

We illustrate this phenomenon by running a simple experiment : using a randomly chosen Markov chain transition matrix of order 5, we sample 1000 sequences of length 5000. Each of them is then used to estimate a Markov model transition matrix of order varying from 2 to 6. For each of these estimates, we have plotted the total variation distance with respect to the generating model (see Figure 1), computed as the quantity $D_{VT}(P,Q) = \sum_{x \in \mathcal{Y}^n} |P(x) - Q(x)|$ for distributions P and Q. It turns out that the optimal estimation in terms of total variation distance between genuine and estimated distributions is obtained with a model of order 2 whereas the generating model is of order 5.

Mixture Transition Distributions aim at providing a model accounting for the number of occurrences of m-letter words, while avoiding the exponential increase with respect to m of the full Markov model parameter's dimension (See

Table 1 for a comparison of the models dimensions).

MTD modeling Let $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ be a sequence of random variables taking values in the finite set $\mathcal{Y} = \{1, \ldots, q\}$. We use the notation,

$$\boldsymbol{Y}_{t_1}^{t_2} = (Y_{t_1}, Y_{t_1+1}, \ldots, Y_{t_2})$$

to refer to the subsequence of the $t_2 - t_1 + 1$ successive variables.

Definition 1 *The random sequence $\boldsymbol{Y}$ is said to be an m^{th} order MTD sequence if*

$$\begin{aligned} \forall t > m, \forall y_1, \ldots, y_t \in \mathcal{Y}, \quad \mathbb{P}(Y_t = y_t | \boldsymbol{Y}_1^{t-1} = \boldsymbol{y}_1^{t-1}) &= \sum_{g=1}^{m} \varphi_g \; \mathbb{P}(Y_t = y_t | Y_{t-g} = y_{t-g}) \\ &= \sum_{g=1}^{m} \varphi_g \boldsymbol{\pi}_g(y_{t-g}, y_t). \end{aligned} \tag{1}$$

where the vector $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_m)$ is subject to the constraints:

$$\forall g \in \{1, \ldots, m\}, \; \varphi_g \geq 0, \tag{2}$$

$$\sum_{g=1}^{m} \varphi_g = 1. \tag{3}$$

and the matrices $\{\boldsymbol{\pi_g} = [\mathbb{P}(Y_t = j | Y_{t-g} = i)]_{1 \leq i,j \leq q}; \; 1 \leq g \leq m\}$ are $q \times q$ stochastic matrices.

A mth-order MTD model is thus defined by a vector parameter,

$$\boldsymbol{\theta} = \Big(\varphi_g, (\boldsymbol{\pi}_g(i,j))_{1 \leq i,j \leq q}\Big)_{1 \leq g \leq m}$$

which belongs to the space

$$\begin{aligned} \Theta \quad = \quad \Bigg\{ \boldsymbol{\theta} \,|\, \forall\, 1 \leq g \leq m,\, 0 \leq \varphi_g \leq 1 \,;\, \sum_{g=1}^{m} \varphi_g = 1 \,; \\ \forall 1 \leq i,j \leq q,\, 0 \leq \boldsymbol{\pi}_g(i,j) \leq 1 \,\text{and}\, \sum_{j=1}^{q} \boldsymbol{\pi}_g(i,j) = 1 \Bigg\}. \end{aligned}$$

It is obvious from the first equality in equation (1) that the MTD model fulfills the Markov property. Thus, MTD models are Markov models with the particularity that each lag $Y_{t-1}, Y_{t-2}, \ldots$ contributes additively to the distribution of the random variable Y_t. Berchtold and Raftery [2] published a complete review of the MTD model. They recall theoretical results on the limiting behavior of the model and on its auto-correlation structure. Details are given about several extensions of this model, such as infinite-lag models, or infinite countable and continuous state space.

We have to point out that Raftery [1] defined the original model with the same transition matrix $\boldsymbol{\pi}$ for each lag $\{Y_{t-g}\}_{g=1,\ldots,m}$. In the sequel, we refer to this model as the *single* matrix MTD model. Later, Berchtold [4] introduced

a more general definition of the MTD models as a mixture of transitions from different subsets of lagged variables $\{Y_{t-m}, \ldots, Y_{t-1}\}$ to the present one Y_t, possibly dropping out some of the lagged dependencies. In this paper, we focus on a slightly more restricted model having a specific but same order transition matrix $\boldsymbol{\pi}_g$ for each lag Y_{t-g}. We denote by MTD_l the MTD model which has a l-order transition matrix for each lag (Definition 2). From now on, the MTD model defined by (1) is denoted accordingly by MTD_1.

Definition 2 *The random sequence $\boldsymbol{Y}$ is a m-order MTD_l sequence if, for all $l, m \in \mathbb{N}$ such that $l < m$, and all $\boldsymbol{y}_1^t \in \mathcal{Y}^t$:*

$$\begin{aligned}
\mathbb{P}(Y_t = y_t | \boldsymbol{Y}_1^{t-1} = \boldsymbol{y}_1^{t-1}) &= \mathbb{P}(Y_t = i_t | \boldsymbol{Y}_{t-1}^{t-m} = \boldsymbol{y}_{t-1}^{t-m}) \\
&= \sum_{g=1}^{m-l+1} \varphi_g \, \mathbb{P}(Y_t = y_t | \boldsymbol{Y}_{t-g-l+1}^{t-g} = \boldsymbol{y}_{t-g-l+1}^{t-g}) \\
&= \sum_{g=1}^{m-l+1} \varphi_g \, \boldsymbol{\pi}_g(\boldsymbol{y}_{t-g-l+1}^{t-g}, y_t).
\end{aligned}$$

holds, where $\boldsymbol{\pi}_g$ is a $q^l \times q$ transition matrix.

Tradeoff between dimension and maximal likelihood Even though MTD models involve a restricted amount of parameters compared to Markov chains, increasing the order l of the model may result in a decreased accuracy of the maximum likelihood estimation. The quality of the trade-off between goodness-of-fit and generalization error a model achieves can be assessed against classical model selection criteria, such as the Bayesian Information Criterion (see illustrations in section 4.2).

However, computing the BIC requires the knowledge of the dimension of the model. This dimension is usually computed as the dimension of the parameter space for a bijective parameterization. In the specific case of the MTD models, the original single-matrix model is parameterized in a bijective way, whereas its generalized version with specific transition matrices for each lag is over-parameterized: in appendix A is given an example of two distinct values of the parameters (φ, ϕ), which both define the same MTD_1 distribution. The dimension of the model is thus lower than the dimension of the parameter space, and computing the BIC using the parameter space dimension would over-penalize the models. A tighter upper bound of the dimension of the MTD_l model is derived in section 2, a bound which is used later to compute the BIC.

The question of estimation As a counterpart for their parsimony, MTD models are difficult to estimate due to the kind of contraints that the estimated transition probabilities $\{\hat{\mathbb{P}}(i_m \ldots i_1; i_0);\ 1 \leq i_m, \ldots, i_0 \leq q\}$ have to comply to. There is indeed no analytical solution to the maximization of the log-likelihood $L_y(\theta) = \mathbb{P}_\theta(Y = y)$ of the MTD models under the constraints the vector φ and the stochastic matrices $\boldsymbol{\pi}_g$ have to fulfill. For a given sequence $\boldsymbol{y} = y_1, \ldots, y_n$ of length n, we recall that the loglikelihood of the sequence $\boldsymbol{y}$ under the MTD_1

model writes

$$\begin{aligned} L_y(\theta) &= \log \mathbb{P}_\theta(\boldsymbol{Y}_1^n = \boldsymbol{y}_1^n) \\ &= \log \left\{ \mathbb{P}(\boldsymbol{Y}_1^m = \boldsymbol{y}_1^m) \prod_{t=m+1}^{n} \left(\sum_{g=1}^{m} \varphi_g \boldsymbol{\pi}_g(y_{t-g}, y_t) \right) \right\}. \end{aligned}$$

The estimation of the original single matrix MTD model already aroused a lot of interest. Although any distribution from this model is defined by a unique parameter $\boldsymbol{\theta}$, the maximum likelihood can not be analytically determined. Li and Kwok [5] propose an interesting alternative to the maximum likelihood with a minimum chi-square method. Nevertheless, they carry out estimations using a non-linear optimization algorithm that is not explicitly described. Raftery and Tavaré [6] obtain approximations of both maximum likelihood and minimum chi-square estimates with numerical procedures from the NAG library which is not freely available. They also show that the MTD model can be estimated using GLIM (Generalized Linear Interactive Modeling) in the specific case where the state space's size q equals 2. Finally, Berchtold [3] developed an ad hoc iterative method implementing a constrained gradient descent optimization. This algorithm is based on the assumption that the vector φ and each row of the matrix $\boldsymbol{\pi}$ are independent. It consists in successively updating each of these vectors constrained to have a sum of components equal to 1 as follows.

Algorithm 1
- *compute partial derivatives of the log likelihood according to each element of the vector*
- *choose a value δ in $[0, 1]$*
- *add δ to the the component with the largest derivative, and subtract δ from the one with the smallest derivative.*

This algorithm has been shown to perform at least better than the previous methods, and it can be extended to the case of the MTD_l models. In this latter case, it estimates *one* of the parameter vectors $\{(\varphi_g, \pi_g); 1 \leq g \leq m\}$ describing the maximum-likelihood MTD distribution. Nevertheless, the choice of the alteration parameter δ remains an issue of the method.

We propose to approximate the maximum likelihood estimate of the MTD model $\left\{ \hat{\mathbb{P}}_{ML}(i_m \dots i_1; i_0); 1 \leq i_m, \dots, i_0 \leq q \right\}$ by coming down to a better known problem: estimation of incomplete data with an Expectation-Maximization (EM) algorithm [7]. We introduce a simple estimation method which allows to approximate *one* parameter vector $\theta = \{(\varphi_g, \pi_g);\ 1 \leq g \leq m\}$ maximizing the log-likelihood.

2 Upper bound of the MTD model dimension

The MTD_1 model over-parametrized. We provide an example of two distinct parameter values (φ, π) defining the same 2nd-order MTD_1 model in appendix A. Moreover, we propose a new parameter set whose dimension is lower. It stems from the straightforward remark that the mth-order MTD_1 model satisfies the following proposition :

Proposition 1 *Transition probabilities of a mth-order MTD$_1$ model satisfy:*

$$\forall i_m, ..., i_g, ..., i_0, i'_g \in \mathcal{Y},$$
$$\mathbb{P}(i_m...i_g...i_1; i_0) - \mathbb{P}(i_m...i'_g...i_1; i_0) = \varphi_g \left[\pi_g(i_g, i_0) - \pi_g(i'_g, i_0)\right]. \quad (4)$$

This simply means that the left-hand side of equation (4) only depends on the parameter components associated to lag g.

Consider a given distribution from MTD$_1$ with parameter $(\varphi_g, \pi_g)_{1\leq g\leq m}$, and let u be an arbitrary element of $\mathcal{Y}$. Each transition probability $\mathbb{P}(i_m...i_1; i_0)$ may be written :

$$\mathbb{P}(i_m...i_1; i_0) = \sum_{g=1}^{m} \varphi_g \left[\pi_g(i_g, i_0) - \pi_g(u, i_0)\right] + \sum_{g=1}^{m} \varphi_g \pi_g(u, i_0). \quad (5)$$

From Proposition 1, it follows that each term of the first sum $\varphi_g \left[\pi_g(i_g, i_0) - \pi_g(u, i_0)\right]$ equals the difference of probabilities $\mathbb{P}(u...ui_gu...u; i_0) - \mathbb{P}(u...u; i_0)$. The second sum is trivially the transition probability from the word $u \dots u$ to i_0.

Let us denote the transition probabilities from m-letters words to the letter j, restricting to words differing from $u \dots u$ by at most one letter :

$$p_u(g\ ; i, j) := \mathbb{P}(u...uiu...u; j), \quad (6)$$

where $u...uiu...u$ is the m-letters word whose letter in position g (from right to left) is i. The quantities in (6) are sufficient to describe the model, as stated in the following proposition.

Proposition 2 *The transition probabilities of a mth-order MTD$_1$ model satisfy:*

$$\forall u \in \mathcal{Y},\ \forall i_m, ..., i_g, ..., i_0 \in \mathcal{Y},$$
$$\mathbb{P}(i_m, ..., i_1; i_0) = \textstyle\sum_{g=1}^{m} \left[p_u(g\ ; i_g, i_0)\ -\ \frac{m-1}{m}\ p_u(i_0)\right].$$

where $p_u(j)$ denotes the value of $p_u(g; u, j)$, whatever the value of g.

For any arbitrary u element of $\mathcal{Y}$, a MTD$_1$ distribution can be parameterized by a vector θ_u from the $(q-1)[1+m(q-1)]$-dimensional set $\bar{\Theta}_u$,

$$\bar{\Theta}_u = \left\{ ((p_u(g; i, j))_{1\leq g\leq m, i,j\in\mathcal{Y}} \text{ such that } \forall g \in \{1, \dots, m\}, \forall i \in \mathcal{Y}, \right.$$
$$\left. \sum_{j\in\mathcal{Y}} p_u(g; i, j) = 1 \text{ and } \forall g, g' \in \{1, \dots, m\}, p_u(g; u, j) = p_u(g'; u, j) \right\} \quad (7)$$

Note that not all $\boldsymbol{\theta}_u$ in $\bar{\Theta}_u$ define a MTD$_1$ distribution: the sum $\sum_{g=1}^{m} p_u(g; i_g, i_0) - \frac{m-1}{m} p_u(i_0)$ may indeed fall outside the interval $[0, 1]$. For this reason, we can only claim that some subset Θ_u of $\bar{\Theta}_u$ is a parameter space for the MTD$_1$ model. However, as the components of a parameter $\theta_u \in \Theta_u$ are transition probabilities, two different parameter values can not define the same MTD distribution. The mapping of Θ_u on the MTD$_1$ model is thus bijective, which results in the dimension of $\bar{\Theta}_u$ being an upper bound of the dimension of the MTD model.

Table 1: **Number of independent parameters required to describe full Markov and MTD$_l$ models (state space size: $q = 4$).** Except for the single matrix MTD model, MTD models originally defined with parameters (φ, π) are over parametrized: the parameter θ_u^l, introduced in section 2, requires far less independent parameters. Note that the 1^{st} order MTD$_1$ (resp. 2^{nd} order MTD$_2$) is equivalent to the 1^{st} order (resp. 2^{nd} order) full Markov model.

	Full	MTD$_1$		MTD$_2$	
Order m	Markov	$\lvert(\varphi,\pi)\rvert$	$\lvert\theta_u^1\rvert$	$\lvert(\varphi,\pi)\rvert$	$\lvert\theta_u^2\rvert$
1	12	12	12		
2	48	25	21	48	48
3	192	38	30	97	84
4	768	51	39	146	120
5	3 072	64	48	195	156

Whereas the original definition of the MTD$_1$ model (1) involves an $m-1+mq(q-1)$-dimensional parameter set, this new parameterization lies in a smaller dimensional space, dropping $q(m-1)$ parameters.

Equivalent parametrization can be set for MTD models having higher order transition matrix for each lag. For any $l \geq 1$, a MTD$_l$ model can be described by a vector composed of the transition probabilities $p_u^l(g; i_l...i_1, j) = \mathbb{P}(u...ui_l...i_1u...u; j)$ for all l-letter words $i_l...i_1$. Denoting by Θ_u^l the corresponding parameter space, its dimension $|\Theta_u^l| = \sum_{k=2}^{l}[q^{k-2}(q-1)^3(m-k+1)] + (1+m(q-1))(q-1)$ is again much smaller than the number of parameters originally required to describe the MTD$_l$ model (see [8], section 2.2, for the counting details). A comparison of the dimensions according to both parametrizations appears in Table 1. We will now make use of the upper bound $|\theta_u^l|$ of the model's dimension to penalize the likelihood in the assessment of MTD models goodness-of-fit (see section 4.2).

3 Estimation

In this section, we expose our estimation method of the MTD$_1$ model (1) having a specific 1^{st} order transition matrix for each lag. The method can easily be adapted for single matrix MTD models as well as for MTD models having different types of transition matrix for each lag. Detailed derivations of the formulas for identical matrix MTD and MTD$_l$ models are presented in appendix B.

To estimate the transition probabilities $\{\mathbb{P}(i_m....i_1; i_0); 1 \leq i_m, ..., i_0 \leq q\}$ of a mth-order MTD$_1$ model, we propose to compute an approximation of *one* set of parameters $\theta = (\varphi_g, \pi_g)_{1 \leq g \leq m}$ which maximizes the likelihood.

3.1 Introduction of a hidden process

Our approach lies on a particular interpretation of the model. The definition of the MTD$_1$ model (1) is equivalent to a mixture of m hidden models where the random variable Y_t is predicted by one of the m Markov chains π_g with the corresponding probability φ_g. Indeed, the coefficients $(\varphi_g)_{g=1,..,m}$ define a

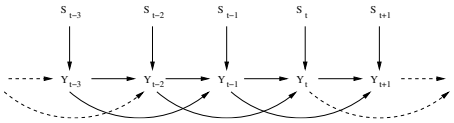


Figure 2: DAG dependency structure of a 2^{nd} order MTD_1 model.

probability measure on the finite set $\{1, ..., m\}$ since they satisfy the constraints (2) and (3).

From now on, we consider a hidden state process $S_1, ..., S_n$ that determines the way according to which the prediction is carried out. The hidden state variables $\{S_t\}$, taking values in the finite set $\mathcal{S} = \{1, ..., m\}$, are independent and identically distributed, with distribution

$$\forall t \leq n, \forall g \in \mathcal{S}, \quad \mathbb{P}(S_t = g) = \varphi_g.$$

The MTD_1 model is then defined as a hidden variable model. The observed variable Y_t depends on the current hidden state S_t and on the m previous variables $Y_{t-1}, ..., Y_{t-m}$. The hidden value at one position indicates which of those previous variables of transition matrices are to be used to draw the current letter : conditionally on the state S_t, the random variable Y_t only depends on teh variable Y_{t-S_t} :

$$\forall t > m, \forall g \in \mathcal{S}, \quad \mathbb{P}(Y_t = y_t | Y_{t-m}^{t-1} = y_{t-m}^{t-1}, S_t = g) = \pi_g(y_{t-g}, y_t).$$

This dependency structure of the model is represented as a Directed Acyclic Graph (DAG) in Figure 2.

So we carry out estimation in the MTD_1 models as estimation in a mixture model where the components of the mixture are m Markov chains, each one predicting the variable Y_t from one of the m previous variables.

3.2 EM algorithm

By considering a hidden variables model, we want to compute the maximum likelihood estimate from incomplete data. The EM algorithm introduced by Dempster et al. [7] is a very classical framework for achieving such a task. It has proved to be particularly efficient at estimating various classes of hidden variable models. We make it entirely explicit in the particular case of the MTD models.

The purpose of the EM algorithm is to approximate the maximum of the log-likelihood of the incomplete data $L_y(\theta) = \log \mathbb{P}_\theta(Y = y)$ over $\theta \in \Theta$ using the relationship

$$\forall \theta, \theta' \in \Theta, L_y(\theta) = Q(\theta|\theta') - H(\theta|\theta')$$

where the quantities Q and H are defined as follows :

$$\begin{aligned} Q(\theta|\theta') &= \mathbb{E}\left[\log \mathbb{P}_\theta(Y, S)|Y = y, \theta'\right] \\ H(\theta|\theta') &= \mathbb{E}\left[\log \mathbb{P}_\theta(Y, S|Y = y)|y, \theta'\right] \end{aligned}$$

The EM algorithm is divided in two steps: E-step (Expectation) and M-step (Maximization). Both steps consist in, respectively, computing and maximizing

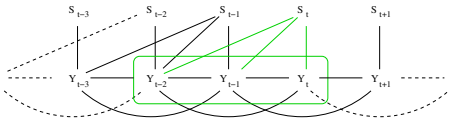


Figure 3: Moral graph of a 2^{nd} order MTD_1 model.

the function $Q(\theta|\theta^{(k)})$, that is the log-likelihood of the complete model conditional on the observed sequence y and on the current parameter $\theta^{(k)}$. Using the fact that the function $\theta \to H(\theta|\theta^{(k)})$ is maximal in $\theta^{(k)}$, Dempster et al. proved that this procedure necessary increases the log-likelihood $L_y(\theta)$. See [9] for a detailed study of the convergence properties of the EM algorithm.

We now derive analytical expressions for both types of updates of the parameters. In this particular case, the log-likelihood of the complete data (Y_{m+1}^n, S_{m+1}^n) conditional on the first m observations Y_1^m writes

$$\log \mathbb{P}_\theta(Y_{m+1}^n, S_{m+1}^n|Y_1^m) = \sum_{t=m+1}^{n} \sum_{g=1}^{m} \sum_{i=1}^{q} \sum_{j=1}^{q} \mathbb{1}_{\{Y_{t-g}=i, Y_t=j, S_t=g\}} \log \pi_g(i,j) + \sum_{t=m+1}^{n} \sum_{g=1}^{m} \mathbb{1}_{\{S_t=g\}} \log \varphi_g. \quad (8)$$

E-step The Estimation step consists in computing the expectation of this function (8) conditional on the observed data y and the current parameter $\theta^{(k)}$, that is calculating, for all $t > m$ and for all element g in $\{1, ..., m\}$, the following quantity,

$$\mathbb{E}(\mathbb{1}_{\{S_t=g\}}|y, \theta^{(k)}) = \mathbb{P}(S_t = g|y, \theta^{(k)}). \quad (9)$$

Then, function Q writes:

$$Q(\theta|\theta^{(k)}) = \sum_{t=m+1}^{n} \sum_{g=1}^{m} \sum_{i=1}^{q} \sum_{j=1}^{q} \left[\mathbb{P}(S_t = g|y, \theta^{(k)}) \log \pi_g(i,j)\right] \mathbb{1}_{\{y_{t-g}=i, y_t=j\}} + \sum_{t=m+1}^{n} \sum_{g=1}^{m} \mathbb{P}(S_t = g|y, \theta^{(k)}) \log \varphi_g. \quad (10)$$

So E-step reduces to computing the probabilities (9), for which we derive an explicit expression by using the theory of graphical models in the particular case of DAG structured dependencies. [10]. First, remark that the state variable S_t depends on the sequence Y only through the $m+1$ variables $\{Y_{t-m}, ..., Y_{t-1}, Y_t\}$:

$$\forall t \leq n, \forall g \in \{1, ..., m\}, \quad \mathbb{P}(S_t = g|y, \theta) = \mathbb{P}(S_t = g|Y_{t-m}^t = y_{t-m}^t, \theta). \quad (11)$$

Indeed, independence properties can be derived from the moral graph (Fig. 3) which is obtained from the DAG structure of the dependencies (Fig. 2) by "marrying" the parents, that is adding an edge between common parents of a variable, and deleting directions. In this moral graph, the set $\{Y_{t-m}, ..., Y_t\}$ separates the variable S_t from the rest of the sequence $\{Y_1, ..., Y_{t-m-1}\}$ so that applying corollary 3.23 from [10] yields:

$$S_t \perp\!\!\!\perp (Y_1^{t-m-1}, Y_{t+1}^n) \mid Y_{t-m}^t$$

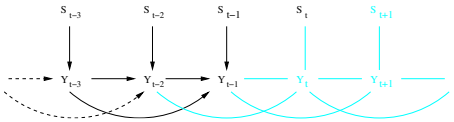


Figure 4: In black: graph of the smallest ancestral set containing S_t and the 2 variables (Y_{t-2}, Y_{t-1}) in the particular case of a 2^{nd} order MTD_1 model. (The part of the structure dependency DAG that is excluded from the smallest ancestral set appears here in light blue.)

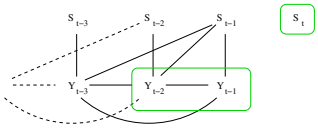


Figure 5: Moral graph of the smallest ancestral set in Figure 4. There is no path between S_t and the subset of 2 variables $\{Y_{t-2}, Y_{t-1}\}$.

From now on, we denote $i_m^0 = i_m i_{m-1} ... i_1 i_0$ any $(m+1)$-letters word composed of elements of $\mathcal{Y}$. For all g in $\{1, ..., m\}$, for all i_m^0 elements of $\mathcal{Y}$, Bayes' Theorem gives:

$$\begin{aligned}
\mathbb{P}(\quad S_t \quad &= g | Y_{t-m}^t = i_m^0, \theta) \\
&= \frac{\mathbb{P}(S_t = g, Y_t = i_0 | Y_{t-m}^{t-1} = i_m^1, \theta)}{\mathbb{P}(Y_t = i_0 | Y_{t-m}^{t-1} = i_m^1, \theta)} \\
&= \frac{\mathbb{P}(Y_t = i_0 | S_t = g, Y_{t-m}^{t-1} = i_m^1, \theta) \mathbb{P}(S_t = g | Y_{t-m}^{t-1} = i_m^1, \theta)}{\sum_{l=1}^{m} \mathbb{P}(Y_t = i_0 | S_t = l, Y_{t-m}^{t-1} = i_m^1, \theta) \mathbb{P}(S_t = l | Y_{t-m}^{t-1} = i_m^1, \theta)}. \quad (12)
\end{aligned}$$

We show below that the probabilities $\mathbb{P}(Y_t = i_0 | S_t = g, Y_{t-m}^{t-1} = i_m^1, \theta)$ and $\mathbb{P}(S_t = g | Y_{t-m}^{t-1} = i_m^1, \theta)$ in expression (12) are entirely explicit. First, conditionnal on θ, the state S_t and the variables Y_{t-m}^{t-1}, the distribution of Y_t writes:

$$\mathbb{P}(Y_t = i_0 | S_t = g, Y_{t-m}^{t-1} = i_m^1, \theta) = \pi_g(i_g, i_0).$$

Second, although the state S_t depends on the $(m+1)$-letters word Y_{t-m}^t, which represents the transition from Y_{t-m}^{t-1} to Y_t observed at time t, it does not depend on the m-letters word formed by only the variables Y_{t-m}^{t-1}. This again follows from the same corollary in [10]. The independence of the variables S_t and Y_{t-m}^{t-1} is derived from the graph of the smallest ancestral set containing these variables, that is the subgraph containing S_t, Y_{t-1}^{t-m} and the whole line of their ancestors (see Figure 4). It turns out that, when considering the moralization of this subgraph (Figure 5), there is no path between S_t and the set Y_{t-m}^{t-1}. This establishes $S_t \perp\!\!\!\perp Y_{t-m}^{t-1}$ and we have

$$\mathbb{P}(S_t = g | Y_{t-m}^{t-1} = i_m^1, \theta) = \mathbb{P}(S_t = g | \theta) = \varphi_g.$$

Finally, the probability (12), is entirely determined by the current parameter θ and does not depend on the time t.

As a result, the k^{th} iteration of Estimation-step consists in calculating, for all g in $\{1, ..., m\}$ and for all $m+1$-letters word i_m^0 of elements of $\mathcal{Y}$,

$$\forall g \in \{1, ..., m\}, \forall\ i_m, ..., i_1, i_0 \in \{1, ..., q\},$$

$$\mathbb{P}_S^{(k)}(g|i_m^0) = \mathbb{P}(S_t = g|Y_{t-m}^t = i_m^0, \theta^{(k)}) = \frac{\varphi_g^{(k)}\pi_g^{(k)}(i_g, i_0)}{\sum_{l=1}^m \varphi_l^{(k)}\pi_l^{(k)}(i_l, i_0)}. \quad (13)$$

M-Step Maximization of the function $Q(\theta|\theta^{(k)})$ with respect to the constraints imposed on the vector φ and on the elements of the transition matrices $\pi_1, ..., \pi_m$ is easily achieved using Lagrange method: $\forall g \in \{1, ..., m\}, \forall i, j \in \{1, ..., q\}$,

$$\varphi_g^{(k+1)} = \frac{1}{n-m} \sum_{i_m ... i_0} \mathbb{P}^{(k)}(g|i_m^0) N(i_m^0) \quad (14)$$

$$\pi_g^{(k+1)}(i,j) = \frac{\sum_{i_m ... i_{g+1} i_{g-1} ... i_1} \mathbb{P}^{(k)}(g|i_m^{g+1} i i_{g-1}^1 j) N(i_m^{g+1} i i_{g-1}^1 j)}{\sum_{i_m ... i_{g+1} i_{g-1} ... i_1 i_0} \mathbb{P}^{(k)}(g|i_m^{g+1} i i_{g-1}^0) N(i_m^{g+1} i i_{g-1}^0)} \quad (15)$$

where sums are carried out for the variables $i_m, ..., i_{g+1}, i_{g-1}, ..., i_1, i_0$ varying from 1 to q, n is the length of the observed sequence y and $N(i_m^0)$ the number of occurrences of the word i_m^0 in this sequence.

Initialization To maximize the chance of reaching the global maximum, we run the algorithm from various starting points. One initialization is derived from contengency tables between each lag y_{t-g} and the present y_t as proposed by Berchtold [3] and several others are randomly drawn from the uniform distribution.

EM-Algorithm for MTD models

- compute the number of occurrences of each $(m+1)$-letters word $N(i_m^0)$
- initialize parameters $(\varphi^{(0)}, \pi^{(0)})$
- choose a stopping rule, *ie* an upper threshold ε on the increase of the log-likelihood
- iterate E and M steps given by equations (13,14,15)
- stop when $L_y(\theta^{(k+1)}) - L_y(\theta^{(k)}) < \varepsilon$

A software implementation of our algorithm is available in the library seq++ at `http://stat.genopole.cnrs.fr/seqpp`.

4 Applications

4.1 Comparison with Berchtold's Estimation

In this paper, we focus on estimation of the MTD_l model (see Definition 2) which has a specific but same order matrix transition for each lag. We evaluate the

Table 2: Maximum log-likelihood of MTD_1 models estimated by EM and Berchtold's algorithm.

Order m	$q = \lvert\mathcal{Y}\rvert$	Berchtold	EM	Sequence
2	3	-486.4	-481.8	Pewee
	45	-1720.1	-1718.5	αA-Crystallin
3	3	-484.0	-480.0	Pewee
	4	-1710.6	-1707.9	αA-Crystallin

performance of the EM algorithm with comparison to the last and best algorithm to date, developed by Berchtold [3]. Among others, Berchtold estimates the parameters of MTD_l models on two sequences analyzed in previous articles: a time serie of the twilight song of the wood pewee and the mouse αA-Crystallin Gene sequence (the complete sequences appear in [6], Tables 7 and 12). The song of the wood pewee is a sequence composed of 3 distinct phrases (refered to as $1, 2, 3$), whereas the αA-Crystallin Gene is composed of 4 nucleotides: a, c, g, t.

We apply our estimation method to these sequences and obtain comparable or higher value of the log-likelihood for both (see Tab. 2). Since the original parameterization of the MTD_1 model is not injective, it is not reasonable to compare their values. To overcome this problem, we computed the parameters defined in (7). The estimated parameters of the 2^{nd} order MTD_1 model on the song of wood Pewee (first line of the Table 2) are exposed in Figure 6 (see appendix C for complete results, that is parameters $\hat{\varphi}, \hat{\pi}_1, \hat{\pi}_2$ used for the estimation procedure and the corresponding full 2^{nd} order transition matrices $\hat{\Pi}$).

For both sequences under study, Pewee and αA-crystallin, EM and Berchtold algorithms lead to comparable estimations. The EM algorithm proves here to be an effective method to maximize the log-likelihood of MTD models. Nevertheless, EM algorithm offers the avantage to be very easy to use. Whereas Berchtold's algorithm requires to set and update a parameter δ to alter the vector φ and each row of the matrices π_g, running the EM algorithm only requires the choice of the threshold ε in the stopping rule.

4.2 Estimation on DNA coding sequences

DNA coding regions are translated into proteins with respect to the genetic code, which is defined on blocks of three nucleotides called codons. Hence, the nucleotides in these regions are constrained in different ways according to their position in the codon. It is common in bioinformatics to use three different transition matrices to predict the nucleotides in the three positions of the codons. This model is called the phased Markov model.

Since we aim at comparing the goodness-of-fit of models with different dimensions, each one's quality is assessed by the maximal value of a penalized likelihood function overs its parameter space. Here we use the Bayesian Information criterion [], defined as :

$$BIC(\mathcal{M}) = -2L_y(\hat{\theta}_{\mathcal{M}}) + d(\mathcal{M}) \log n$$

Figure 6: Estimation of a 2^{nd} order MTD_1 model on the song of the wood pewee. We use u=1 (song n°1) as reference letter to express the parameters defined in (7).

Estimates obainted with:

- Berchtold's algorithm ($L_y(\hat{\theta}) = -486.4$):

$$[\hat{p}_1(1;i,j)]_{1\leq i,j\leq 3} = \begin{pmatrix} 0.754169 & 0.198791 & 0.073356 \\ 0.991696 & 0. & 0.03462 \\ 0.993579 & 0.003497 & 0.02924 \end{pmatrix}$$

$$[\hat{p}_1(2;i,j)]_{1\leq i,j\leq 3} = \begin{pmatrix} 0.754169 & 0.198791 & 0.073356 \\ 0.137205 & 0.213411 & 0.649384 \\ 0.048023 & 0.927598 & 0.044116 \end{pmatrix}$$

- EM-algorithm ($L_y(\hat{\theta}) = -481.8$):

$$[\hat{p}_1(1;i,j)]_{1\leq i,j\leq 3} = \begin{pmatrix} 0.75305 & 0.200475 & 0.046475 \\ 0.991475 & 0. & 0.008525 \\ 0.996425 & 0.003575 & 0. \end{pmatrix}$$

$$[\hat{p}_1(2;i,j)]_{1\leq i,j\leq 3} = \begin{pmatrix} 0.75305 & 0.200475 & 0.046475 \\ 0.137525 & 0.21135 & 0.651125 \\ 0.02805 & 0.925475 & 0.046475 \end{pmatrix}$$

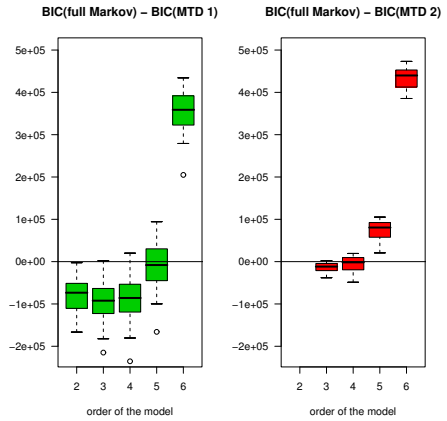


Figure 7: Difference according to the BIC criterion between MTD models and the corresponding fully parametrized Markov Model.

where $\hat{\theta}_{\mathcal{M}}$ stands for the maximum likelihood estimate of model $\mathcal{M}$. The lower the BIC a model achieves, the more pertinent it is.

BIC evaluation has been computed on DNA coding sequence sets from bacterial genomes. Each of these sequence sets has length ranging from 1 500 000 to 5 000 000. Displayed values in Figure 7 are averages over the 15 sequences set of the difference between the BIC value achieved the full Markov model and the one achieved by the MTD model of the same order. Whenever this figure is positive, the MTD model has to be preferred to the full Markov model.

The full Markov model turns out to outperform the MTD model when the order is inferior to 4. This is not surprising since the estimation is computed over large datasets that provide a sufficient amount of information with respect to the number of parameters of the full model. However, the 5^{th} order MTD model and full Markov model have comparable performances, and the MTD model outperforms the full Markov model for higher orders. This is an evidence that although MTD only approximate the full Markov models, their estimation accuracy decreases slower with the order.

Even more striking is the comparison of the MTD_2 model with the full Markov model. Whatever the order of the model, its goodness-of-fit is at least equivalent to the one achieved by the full Markov model. The MTD_l model turns out to be a satisfactory trade-off between dimension and estimation accuracy.

5 Acknowledgements

We thank Bernard Prum and Catherine Matias for their very constructive suggestions, and Vincent Miele for implementing the EM algorithm in the seq++ library.

A Example of equivalent parameters defining the same MTD_1 model

Let the size state space be 4 as for DNA sequences $\mathcal{Y} = \{a, c, g, t\}$ and consider these two 2^{nd} order MTD_1 model parameters θ, θ'.

$$\varphi = (0.3, 0.7) \qquad \pi_1 = \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.4 \\ 0.4 & 0.3 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.2 & 0.4 \\ 0.4 & 0.2 & 0.2 & 0.2 \end{pmatrix} \qquad \pi_2 = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.7 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.3 & 0.3 & 0.3 & 0.1 \\ 0.3 & 0.2 & 0.3 & 0.2 \end{pmatrix}$$

$$\varphi' = (0.2, 0.8) \quad \pi_1' = \begin{pmatrix} 0.2 & 0.1 & 0.2 & 0.5 \\ 0.65 & 0.25 & 0.05 & 0.05 \\ 0.35 & 0.1 & 0.05 & 0.5 \\ 0.65 & 0.1 & 0.05 & 0.2 \end{pmatrix} \quad \pi_2' = \begin{pmatrix} 0.075 & 0.1375 & 0.15 & 0.6375 \\ 0.1625 & 0.225 & 0.4125 & 0.2 \\ 0.25 & 0.3125 & 0.325 & 0.1125 \\ 0.25 & 0.225 & 0.325 & 0.2 \end{pmatrix}$$

Both parameters define the same 2^{nd} order Markov transition matrix Π.

$$\Pi = \begin{array}{c} \\ aa \\ ac \\ ag \\ at \\ ca \\ cc \\ cg \\ ct \\ ga \\ gc \\ gg \\ gt \\ ta \\ tc \\ tg \\ tt \end{array} \begin{array}{c} \begin{array}{cccc} a & c & g & t \end{array} \\ \begin{pmatrix} 0.1 & 0.13 & 0.16 & 0.61 \\ 0.19 & 0.16 & 0.13 & 0.52 \\ 0.13 & 0.13 & 0.13 & 0.61 \\ 0.19 & 0.13 & 0.13 & 0.55 \\ 0.17 & 0.2 & 0.37 & 0.26 \\ 0.26 & 0.23 & 0.34 & 0.17 \\ 0.2 & 0.2 & 0.34 & 0.26 \\ 0.26 & 0.2 & 0.34 & 0.2 \\ 0.24 & 0.27 & 0.3 & 0.19 \\ 0.33 & 0.3 & 0.27 & 0.1 \\ 0.27 & 0.27 & 0.27 & 0.19 \\ 0.33 & 0.27 & 0.27 & 0.13 \\ 0.24 & 0.2 & 0.3 & 0.26 \\ 0.33 & 0.23 & 0.27 & 0.17 \\ 0.27 & 0.2 & 0.27 & 0.26 \\ 0.33 & 0.2 & 0.27 & 0.2 \end{pmatrix} \end{array}$$

B EM algorithm for other MTD models

B.1 Single matrix MTD model: iteration k.

E-Step $\forall g \in \{1, ..., m\}, \forall\, i_m, ..., i_1, i_0 \in \{1, ..., q\}$,

$$\mathbb{P}_S^{(k)}(g|i_m^0) = \frac{\varphi_g^{(k)} \pi^{(k)}(i_g, i_0)}{\sum_{l=1}^{m} \varphi_l^{(k)} \pi^{(k)}(i_l, i_0)}.$$

M-Step $\forall g \in \{1, ..., m\}, \forall i, j \in \{1, ..., q\}$,

$$\varphi_g^{(k+1)} = \frac{1}{n-m}\sum_{i_m\dots i_0}\mathbb{P}^{(k)}(g|i_m^0)N(i_m^0)$$

$$\pi^{(k+1)}(i,j) = \frac{\sum_{g=1}^{m}\sum_{i_m\dots i_{g+1}i_{g-1}\dots i_1}\mathbb{P}^{(k)}(g|i_m^{g+1}ii_{g-1}^1j)N(i_m^{g+1}ii_{g-1}^1j)}{\sum_{g=1}^{m}\sum_{i_m\dots i_{g+1}i_{g-1}\dots i_1i_0}\mathbb{P}^{(k)}(g|i_m^{g+1}ii_{g-1}^0)N(i_m^{g+1}ii_{g-1}^0)}$$

where sums are carried out for the variables $i_m,...,i_{g+1},i_{g-1},...,i_1,i_0$ varying from 1 to q, n is the length of the observed sequence y and $N(i_m^0)$ the number of occurrences of the word i_m^0 in this sequence.

B.2 MTD_l model: iteration k.

E-Step $\forall g\in\{1,...,m-l+1\},\forall i_m,...i_1,i_0\in\{1,...,q\}$,

$$\mathbb{P}_S^{(k)}(g|i_m^0)=\frac{\varphi_g^{(k)}\pi_g^{(k)}(i_{g+l-1}^g,i_0)}{\sum_{h=1}^{m-l+1}\varphi_h^{(k)}\pi_h^{(k)}(i_{h+l-1}^h,i_0)}.$$

M-Step $\forall g\in\{1,...,m\},\forall i_l,...,i_1,j\in\{1,...,q\}$,

$$\varphi_g^{(k+1)} = \frac{1}{n-m}\sum_{u_m\dots u_0}\mathbb{P}_S^{(k)}(g|u_m^0)N(u_m^0)$$

$$\pi_g^{(k+1)}(i_li_{l-1}...i_1,j) = \frac{\sum_{u_m\dots u_{g+l}u_{g-1}\dots u_1}\mathbb{P}_S^{(k)}(g|u_m^{g+l}i_l^1u_{g-1}^1j)N(u_m^{g+l}i_l^1u_{g-1}^1j)}{\sum_{u_m\dots u_{g+l}u_{g-1}\dots u_1u_0}\mathbb{P}_S(g|u_m^{g+l}i_l^1u_{g-1}^0)N(u_m^{g+l}i_l^1u_{g-1}^0)},$$

where sums are carried out for the variables $u_m,...,u_{g+l},u_{g-1},...,u_1,u_0$ varying from 1 to q, n is the length of the observed sequence y and $N(i_m^0)$ the number of occurrences of the word i_m^0 in this sequence.

C 2^{nd} order MTD_1 estimates obtained on the song of wood pewee (Section 4.1).

Berchtold's algorithm:
$(L_y(\hat\theta)=-486.4)$

$$\hat\pi_2=\begin{pmatrix}0.996 & 0 & 0.004\\ 0.152 & 0.020 & 0.828\\ 0.003 & 0.997 & 0\end{pmatrix}$$

$\hat\varphi=(0.269,0.731)$

EM-algorithm:
$(L_y(\hat\theta)=-481.8)$

$$\hat\pi_1=\begin{pmatrix}0.097 & 0.739 & 0.164\\ 0.980 & 0 & 0.020\\ 0.987 & 0.013 & 0\end{pmatrix}$$

$\hat\varphi=(0.275,0.725)$

$$\hat{\pi}_1 = \begin{pmatrix} 0.102 & 0.729 & 0.169 \\ 0.969 & 0 & 0.031 \\ 0.987 & 0.013 & 0 \end{pmatrix} \qquad \hat{\pi}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0.151 & 0.015 & 0.834 \\ 0 & 1 & 0 \end{pmatrix}$$

These estimated parameters define respectively the following 2^{nd} order Markov transition matrices $\hat{\Pi}_B$ and $\hat{\Pi}_{EM}$.

$$\hat{\Pi}_B = \begin{pmatrix} 0.754169 & 0.198791 & 0.073356 \\ 0.991696 & 0. & 0.03462 \\ 0.993579 & 0.003497 & 0.02924 \\ 0.137205 & 0.213411 & 0.649384 \\ 0.374732 & 0.01462 & 0.610648 \\ 0.376615 & 0.018117 & 0.605268 \\ 0.048023 & 0.927598 & 0.044116 \\ 0.28555 & 0.728807 & 0.00538 \\ 0.287433 & 0.732304 & 0. \end{pmatrix} \quad \hat{\Pi}_{EM} = \begin{pmatrix} 0.75305 & 0.200475 & 0.046475 \\ 0.991475 & 0. & 0.008525 \\ 0.996425 & 0.003575 & 0. \\ 0.137525 & 0.21135 & 0.651125 \\ 0.37595 & 0.010875 & 0.613175 \\ 0.3809 & 0.01445 & 0.60465 \\ 0.02805 & 0.925475 & 0.046475 \\ 0.266475 & 0.725 & 0.008525 \\ 0.271425 & 0.728575 & 0. \end{pmatrix}$$

References

[1] Adrian E. Raftery. A model for high-order Markov chains. Journal of the Royal Statistical Society. Series B, 47(3):528–539, 1985.

[2] André Berchtold and Adrian E. Raftery. The mixture transition distribution model for high-order markov chains and non-gaussian time series. Statistical Science, 17:328–356, 2002.

[3] André Berchtold. Estimation in the mixture transition distribution model. Journal of Time Series Analysis, 22(4):379–397, 2001.

[4] André Berchtold. Autoregressive modeling of markov chains. In Statistical Modelling: Proceedings of the 10^{th} International Workshop on Statistical Modelling, pages 19–26. Springer-Verlag, 1995.

[5] W.K. Li and Michael C.O. Kwok. Some results on the estimation of a higher order markov chain. Commun. Stat. Simulat., 19(1):363–380, 1990.

[6] Adrian E. Raftery and Simon Tavaré. Estimation and modelling repeated patterns in high order markov chains with the mixture transition distribution model. Journal of the Royal Statistical Society Applied Statistics, 43(1):179–199, 1994.

[7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B., 39:1–38, 1977.

[8] Adeline Grelot. Estimation Bayésienne d'un modèle de mélange de transition markovienne - MSc Report available at http://stat.genopole.cnrs.fr/publi/...-. 2005.

[9] C. Wu. On the convergence properties of the em algorithm. The Annals of Statistics, 11(1):95–103, 1983.

[10] Steffen L. Lauritzen. Graphical models. Repr. Oxford Statistical Science Series. 17., 1998.

ANNEXE B

Seq++ : analyzing biological sequences with a range of Markov-related models, *Bioinformatics*

Le développement d'outils informatiques permettant la mise en œuvre des méthodes développées pour l'analyse des séquences, en particulier dans le cas de la modélisation des séquences des macro-molécules biologiques, fait partie intégrante de ce travail de thèse.

Le laboratoire Statistique et Génome *fournit, sous forme de logiciels libres, l'ensemble des outils permettant la mise en œuvre des algorithmes qui y sont développés. Ces outils sont regroupés dans une librairie logicielle,* `seq++`*, dont le développement et la maintenance sont assurés principalement par* V. MIELE.

Cette librairie logicielle a fait l'objet d'une publication courte dans Bioinformatics *en 2004.*

BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 11 2005, pages 2783–2784
doi:10.1093/bioinformatics/bti389

Sequence analysis

seq++: analyzing biological sequences with a range of Markov-related models

Vincent Miele*, Pierre-Yves Bourguignon, David Robelin, Grégory Nuel and Hugues Richard

UMR CNRS 8071 Statistique et Génome, 523 place des Terrasses, 91000 Evry, France

Received on October 28, 2004; revised on February 1, 2005; accepted on March 9, 2005
Advance Access publication March 17, 2005

ABSTRACT

Summary: The seq++ package offers a reference set of programs and an extensible library to biologists and developers working on sequence statistics. Its generality arises from the ability to handle sequences described with any alphabet (nucleotides, amino acids, codons and others). seq++ enables sequence modelling with various types of Markov models, including variable length Markov models and the newly developed parsimonious Markov models, all of them potentially phased. Simulation modules are supplied for Monte Carlo methods. Hence, this toolbox allows the study of any biological process which can be described by a series of states taken from a finite set.

Availability: Under the GNU General Public Licence at http://stat.genopole.cnrs.fr/seqpp

Contact: miele@genopole.cnrs.fr

INTRODUCTION

A considerable range of genomic data can be modelled as sequences of characters taken from various alphabets. Obvious candidates are the nucleic acids or protein sequences. However, RNA or protein secondary structures can similarly be described using character strings. This paper presents seq++ , a C++ software library, aiming to be a reference environment for studying the statistical properties of these sequences using a comprehensive set of Markov models. It already implements most of the classical methods, as well as some more recent ones, and offers an extensible framework to experiment with new models, which can be included in subsequent releases of the library.

MODELS AND FEATURES

The key to the flexibility of seq++ is its independence from any given alphabet: the algorithms implemented in seq++ handle strings of tokens. These tokens, which can be several characters long are enumerated in an alphabet file also allowing the definition of synonymous token groups. For instance, studying amino acid composition properties of protein sequences is possible by grouping amino acids according to their physical or chemical properties and assigning them a suitable label.

The algorithms themselves focus on Markov chains (MC) sequence modelling, including phased models. These models can be useful when phased emission heterogeneity occur in the observations. For example, it is well known that the third nucleotide of a codon has different occurrence patterns than the two preceding ones. This can be taken into account by fitting one transition matrix (which is the matrix determining emission probabilities for the observations) per phase to yield more accurate results (Borodovsky *et al*., 1995).

Models provided by seq++ include variable length Markov chains (VLMC) (Bühlmann and Wyner, 1999) and parsimonious Markov chains (PMC) [Bourguignon and Robelin, 2004; P.Y. Bourguignon, 2005 (submitted for publication)]. Let $Y_{i,0<i\leq l}$ be the set of tokens in a sequence of length l modelled by a d-order Markov model. In VLMC, the prediction of Y_i (the token at position i) can be determined by the preceding words of variable length $(Y_{i-d'} \cdots Y_{i-1}, d' \leq d)$. Using words of length less than d can considerably reduce the number of parameters and increase the quality of the model adjustment. The same motivations underlie the use of PMC, where predictors with identical emission probabilities are grouped into motifs representing degenerated token words with possible gaps (Fig. 1). Nevertheless such improvements in the model accuracy can require computational costs.

Therefore, methods are implemented in the library for Markovian transition matrix estimation, stationary distribution calculus, word probabilities, total variation distance between two Markovian matrices, likelihood and Bayesian information criterion (BIC) calculus. The efficiency of eigenproblems computation is ensured by the use of Arnoldi algorithms (Lehoucq *et al*., 1996). These methods aim, for example, to contribute to a motif-detection algorithm (Bulyk, 2003): given a model based on a selection of known motifs of interest and a background model on a target sequence, a sliding window approach can be developed based on the ratio of the likelihoods of the 'site' model and the 'background' model (high scores corresponding to potential sites; Fig. 2).

In addition, the programs `estim_m`, `estim_vlm` and `estim_pm` are also released in seq++ to perform a set of calculus associated with models MC, VLMC and PMC, respectively. Moreover, the program `simul_m` simulates a sequence according to a Markovian matrix (or p matrices for p phases) previously estimated. When working on homogeneous sequences, biologists are often interested in the P-value of an observation. This P-value is frequently impossible to calculate analytically, thus it can be estimated using simulated control sequences. As a result seq++ provides an efficient environment for Monte Carlo methods.

*To whom correspondence should be addressed.

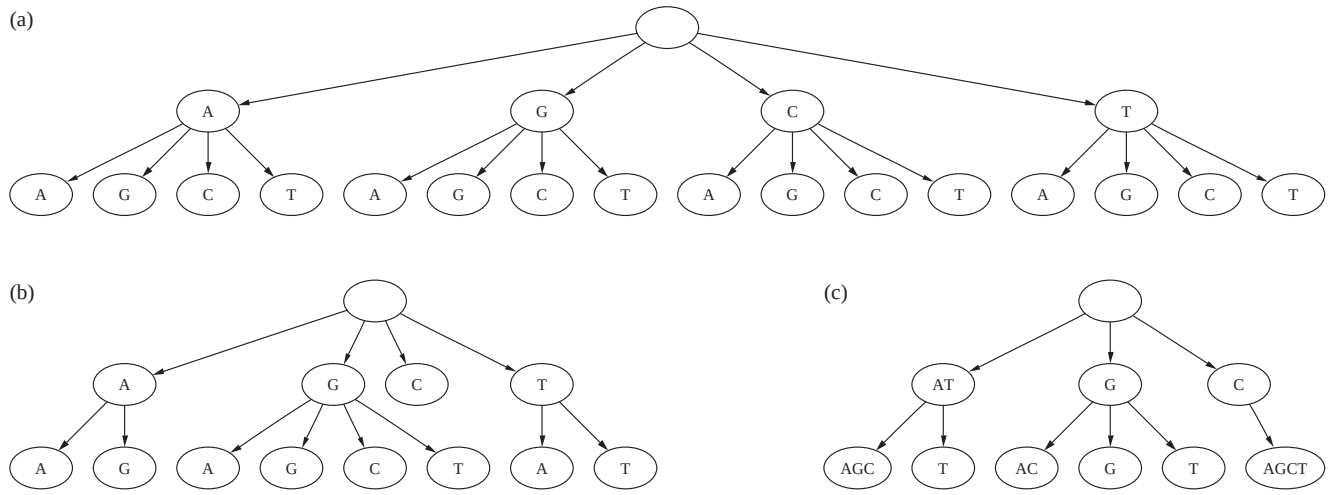


Fig. 1. A Markovian model can be represented by a tree: the root corresponds to the letter Y_t to be predicted, the possible values of Y_{t-1} are on the first level, those of Y_{t-2} on the second and so on. (**a**) represents an MC of order 2. In a VLMC, branches can be cut: as in (**b**), when $Y_{t-1} = C$, the law of Y_t does not depend on Y_{t-2}. In a PMC, subtrees can be merged: as in (**c**) the two trees below $Y_{t-1} = A$ and $Y_{t-1} = T$ generate a unique tree. In each case, the number of predictors is equal to the number of leaves in the tree.

```
ord = 2;
SequenceSet mot("motifs.fna","dna",ord);
nphase = mot(0).tell_length();
PhasedMarkov m_motif(mot, nphase);
SequenceSet set("target.gb","dna",ord);
Sequence & seq = set(0);
Markov  m_backg(seq);
for(t=nphase-1;t<seq.tell_length();t++){
   a = m_motif.proba(seq.t-nphase+1,t);
   b = m_backg.proba(seq.t-nphase+1,t);
   cout<<t<<" score: "<<a/b<<endl;}
```

Fig. 2. Code example for motif detection on DNA, where `m_motif` and `m_backg` are the models of order `ord` estimated on the known motifs and the target sequence, respectively. `proba` returns the probability of the word observed between positions `t-nphase+1` and `t`.

DESIGN AND AVAILABILITY

The object-oriented design of seq++ allows for further evolution. A module dedicated to the Mixture Transition Distribution (MTD) model (Lebre, 2004) is planned for future seq++ releases. To our knowledge, seq++ is the only available library for Markovian sequence analysis. A similar project, libsequence (Thornton, 2003), is dedicated to single nucleotide polymorphism analysis. The GHMM library (http://ghmm.org) for hidden Markov models may be a valuable alternative for various problematics on heterogeneous sequences.

The package is written in ANSI C++ and developed on $\times 86$ GNU/Linux systems with GCC 3.4. It has been successfully tested with Intel ICC 8.0, on Sun systems using GCC 3.3 and Apple Mac OSX systems with GCC 3.1. Compilation and installation are compliant with the GNU standard procedure. The library is free and available at http://stat.genopole.cnrs.fr/seqpp. Online documentation is also available. Software using seq++ (Robelin *et al*., 2003) dedicated to DNA bioinformatics can also be accessed online. seq++ is licensed under the GNU General Public License (http://www.gnu.org/licences.html).

ACKNOWLEDGEMENTS

We are grateful to M.Baudry for the hardware support, M.Hoebeke and P.Nicolas for the programming advice.

REFERENCES

Borodovsky,M. *et al*. (1995) Detection of new genes in a bacterial genome using Markov models for three gene classes. *Nucleic Acids Res.*, **25**, 3554–3562.

Bourguignon,P.Y. and Robelin,D. (2004) Modèles de Markov parcimonieux, *Actes de JOBIM*.

Bühlmann,P. and Wyner,A.J. (1999) Variable length Markov chains. *Ann. Stat.*, **27**, 480–513.

Bulyk,M.L. (2003) Computational prediction of transcription-factor binding site locations, *Genome Biol.*, **5**, 201.

Lebre,S. (2004) Estimation de modèle MTD, *Evry University Internal Report*.

Lehoucq,R.B., Sorensen,B. and Yang,C. (1996) ARPACK user's guide: solution of large-scale eigenvalue problems by implicitly restarted Arnoldi methods, *Rice University Technical Report*.

Robelin,D. *et al*. (2003) SIC: a tool to detect short inverted segments in a biological sequence. *Nucleic Acids Res.*, **31**, 3669–3671.

Thornton,K. (2003) libsequence, a C++ class library for evolutionary genetic analysis. *Bioinformatics*, **19**, 2325–2327.

ANNEXE C

Recherche de points chauds de recombinaison meïotique, *Genome Research*

L'article qui suit résulte d'une collaboration ponctuelle avec une équipe de biologistes commune à l'INRA et au Centre National de Génotypage (CNG), initiée à la demande de cette équipe.

Le projet poursuivi au CNG consistait à caractériser finement la fréquence des événements de recombinaison meïotique dans les différentes régions du chromosome IV de la plante Arabidopsis thaliana. *La recombinaison meïotique est liée au mécanisme de ségrégation des chromosomes lors de la meïose : afin d'assurer qu'au cours de la division cellulaire, une copie de chacun des chromosomes soit bien attribuée à chacune des cellules filles, les cellules eucaryotes induisent une cassure des deux brins de la molécule d'ADN. Pour remédier à ce dommage, la cellule procède alors à l'échange d'un brin d'ADN entre les copies homologues des chromosomes, seule manière de rétablir, pour chaque copie de chaque chromosome, au moins un brin complet à partir duquel rétablir la molécule d'ADN dans son intégralité. Cet échange se traduit par une résistance mécanique lorsque les deux copies homologues d'un chromosome sont tractés par les fuseaux mitotiques vers les deux pôles de la division, et, en l'absence de cette résistance, les fuseaux de délitent. Ainsi, la mitose ne parvient à son terme que si les fuseaux opposés attirent bien une copie de chaque chromosome dans chaque cellule fille.*

Ce faisant, la recombinaison est responsable de la diversification des séquences nucléiques. En effet, les chromosomes résultant d'une recombinaison chez un individu sont composés de séquences héritées de chacun des deux parents de l'individu, et permet par conséquent un brassage des allèles. La compréhension de ce mécanisme est donc une étape essentielle vers l'analyse du déséquilibre de liaison, *c'est-à-dire de la corrélation entre les allèles voisins sur le chromosome chez un individu.*

*Notre implication dans ce projet a été tardive, et provoquée par des difficultés d'interprétation des données générées pour ce projet : l'équipe INRA/CNG avait en effet créé une souche d'*A. thaliana *portant deux copies du chromosome IV issues de souches différentes, et entre lesquelles de nombreux points de polymorphisme étaient recensés. Cette souche a ensuite permis d'engendrer environ un millier de descendants. Chacun de ceux-ci a naturellement reçu un chromosome de la plante mère résultant d'exactement une meïose, et donc un événement de recombinaison. Chacun de ces événements de recombinaison ayant eu lieu indépendamment au cours de la génèse de chaque descendant, le génotypage des plantes filles a permis de recenser le nombre d'événements de recombinaison dans un peu moins d'une centaine d'intervalles flanqués de site polymorphes sur la séquence génomique.*

A l'issue de ce recensement, il était difficile d'identifier les intervalles significativement chauds *(*i.e. *où se concentrent les événements de recombinaison), en particulier car plusieurs phénomènes se superposaient : les intervalles ne sont pas tous de même longueur, et chacune de leurs extrémités n'avaient pas pu être génotypées chez un même nombre de plantes filles pour des raisons expérimentales. Aussi avons-nous développé un modèle statistique binomial simple, sous lequel nous avons calculé la significativité de la déviation du taux de recombinaison observé dans un intervalle par rapport au taux*

moyen le long du chromosome. Ce travail a permis de clarifier amplement les données, et ainsi d'identifier un certain nombre de régions chaudes dans le chromosome IV.

Ce travail a fait l'objet d'une publication dans la revue Genome Research.

Variation in crossing-over rates across chromosome 4 of Arabidopsis thaliana reveals the presence of meiotic recombination "hot spots"

Jan Drouaud, Christine Camilleri, Pierre-Yves Bourguignon, Aurélie Canaguier, Aurélie Bérard, Daniel Vezon, Sandra Giancola, Dominique Brunel, Vincent Colot, Bernard Prum, Hadi Quesneville and Christine Mézard

Genome Res. published online Dec 12, 2005;
Access the most recent version at doi:10.1101/gr.4319006

Letter

Variation in crossing-over rates across chromosome 4 of *Arabidopsis thaliana* reveals the presence of meiotic recombination "hot spots"

Jan Drouaud,[1] Christine Camilleri,[1,2] Pierre-Yves Bourguignon,[3] Aurélie Canaguier,[1,6] Aurélie Bérard,[1,2] Daniel Vezon,[1] Sandra Giancola,[1,2] Dominique Brunel,[1,2] Vincent Colot,[4] Bernard Prum,[3] Hadi Quesneville,[5] and Christine Mézard[1,7]

[1]Station de Génétique et d'Amélioration des Plantes, Institut Jean-Pierre Bourgin, Institut National de la Recherche Agronomique (INRA), 78026, Versailles cedex, France; [2]INRA/CNG, 91057 Evry cedex, France; [3]Laboratoire Statistique et Génome, UMR 8071 Centre National de la Recherche Scientifique (CNRS)-INRA-Université Evry Val d'Essonne, (UEVE), 91000 Evry, France; [4]Unité de Recherche en Génomique Végétale (URGV), INRA/CNRS/UEVE, CP5708, 91057 Evry cedex, France; [5]Laboratoire de Dynamique du Génome et Evolution, Institut Jacques Monod, 75251 Paris cedex 05, France

Crossover (CO) is a key process for the accurate segregation of homologous chromosomes during the first meiotic division. In most eukaryotes, meiotic recombination is not homogeneous along the chromosomes, suggesting a tight control of the location of recombination events. We genotyped 71 single nucleotide polymorphisms (SNPs) covering the entire chromosome 4 of *Arabidopsis thaliana* on 702 F2 plants, representing 1404 meioses and allowing the detection of 1171 COs, to study CO localization in a higher plant. The genetic recombination rates varied along the chromosome from 0 cM/Mb near the centromere to 20 cM/Mb on the short arm next to the NOR region, with a chromosome average of 4.6 cM/Mb. Principal component analysis showed that CO rates negatively correlate with the G+C content ($P = 3 \times 10^{-4}$), in contrast to that reported in other eukaryotes. COs also significantly correlate with the density of single repeats and the CpG ratio, but not with genes, pseudogenes, transposable elements, or dispersed repeats. Chromosome 4 has, on average, 1.6 COs per meiosis, and these COs are subjected to interference. A detailed analysis of several regions having high CO rates revealed "hot spots" of meiotic recombination contained in small fragments of a few kilobases. Both the intensity and the density of these hot spots explain the variation of CO rates along the chromosome.

[Supplemental material is available online at www.genome.org.]

Meiotic crossovers (COs) and sister chromatid cohesion provide physical links between homologous chromosomes ensuring proper chromosome segregation during the first meiotic division. In most eukaryotes, there is always at least one CO per pair of homologs (obligatory crossover) (Jones 1984, 1987). Cytological, genetic, and molecular studies in many organisms have demonstrated that COs are not evenly distributed along the chromosomes (Jones 1987; Carpenter 1988; Lynn et al. 2002). The tight control of the number and/or localization of COs is crucial. Mutations that reduce CO formation increase chromosome nondisjunction in organisms as diverse as *Saccharomyces cerevisiae*, *Schizosaccharomyces pombe*, *Caenorhabditis elegans*, *Drosophila melanogaster* (female), *Arabidopsis thaliana*, and the mouse (for review, see Lynn et al. 2004).

In yeast, the distribution of meiotic recombination events (COs and noncrossover gene conversions; NCOs) along chromosomes has been studied in detail by locating DNA double-strand breaks (DSBs), which initiate meiotic recombination (Baudat and Nicolas 1997; Gerton et al. 2000). These studies showed that DSBs tend to be clustered in chromosomal domains away from telomeres and centromeres (Gerton et al. 2000; Borde et al. 2004). In mammals, COs are also nonrandomly distributed along the chromosomes, with alternate domains having higher or lower levels of recombination (Kong et al. 2002; Nachman 2002). The CO rates tend to be low near the centromeres and increase toward the telomeres. In plants, the CO rates also vary along chromosomes (for review, see Anderson and Stack 2002). In general, centromeric regions have low CO rates compared to telomeric regions. However, in plants, there have been very few high-resolution studies in a single chromosome.

Many sequence parameters have been linked to the variation of CO rates in eukaryotes. In yeast and mammals, several studies have found a correlation between a high G+C content and a high rate of recombination in large domains (Gerton et al. 2000; Fullerton et al. 2001; Yu et al. 2001; Kong et al. 2002; Petes and Merker 2002; Jensen-Seaman et al. 2004). However, within 2–3 kb of the recombination initiation site no correlation between the G+C content and the distribution of COs in both yeast and humans was found (for review, see de Massy 2003) and, second, in human, rat, and mouse, when CpG ratio is included in a multiple regression analysis, correlation with the G+C content becomes negative (Kong et al. 2002; Jensen-Seaman et al. 2004). In wheat, barley, and maize, gene-rich regions are more recombinationally active than gene-poor regions (for review, see

[6]Present address: URGV, INRA/CNRS, 2, rue Gaston Crémieux, CP5708, 91057 Evry cedex, France
[7]Corresponding author.
E-mail mezard@versailles.inra.fr; fax (33) 1 30 83 33 19.
Article published online ahead of print. Article and publication date are at http://www.genome.org/cgi/doi/10.1101/gr.4319006.

Schnable et al. 1998). In humans, female CO rates are not correlated with gene density on chromosome 21 (Lynn et al. 2000) whereas male CO rates are correlated, suggesting a different type of control. There are also conflicting results when correlating the density of transposable elements (TEs) and recombination rates (see Wright et al. 2003). Nevertheless, differences in meiotic CO rates between the sexes have been demonstrated in many higher eukaryotes (Lenormand and Dutheil 2005). Therefore, the primary DNA sequence itself cannot explain all of the variation of meiotic recombination.

In *S. cerevisiae* and *S. pombe*, hot spots have been defined as small DNA fragments of 1–2 kb, centered around meiotic DSBs that are repaired, using the homologous chromosome, to produce COs or NCOs (Keeney 2001). In mice, humans, and plants, several such regions have been studied in detail and have been found to share common features with hot spots described in yeast. These include high level of COs and NCOs clustered in small segments (1–2 kb) and a lack of clear consensus sequences (de Massy 2003; Kauppi et al. 2004; Rafalski and Morgante 2004). The distribution of meiotic hot spots along chromosomes is uneven, which suggests a local control of DSB formation. A lot of effort has been made recently to characterize this fine-scale variation of recombination rates mainly in humans but also in other eukaryotes for various reasons among which is to gain insight into the underlying mechanisms, to assist association studies, or to improve inferences from polymorphism data about selection and population history. However, except for *S. cerevisiae*, only a few regions have been characterized at the molecular level in other eukaryotes and more genome-wide studies are needed to unravel the determinants of hot spot activity.

The availability of the *Arabidopsis* genome sequence (The *Arabidopsis* Genome Initiative 2000) and the recent development of powerful high-throughput genotyping techniques (Gut 2001; Kwok 2001), allow us to determine precisely the location and rates of COs on one chromosome. Here, we show that CO rates are highly variable on chromosome 4 of *Arabidopsis*, with some regions having five times more COs than the chromosome average. The CO rates significantly negatively correlate with the G+C content and also significantly correlate with the density of single repeats and with CpG ratio. However, they do not correlate significantly either with genes, pseudogenes, transposable elements, or dispersed repeats. Our data also confirm that COs are subjected to interference on chromosome 4. Finally, we provide evidence of meiotic recombination hot spots and show that both their activity and density contribute to the variation of the CO rates.

Results

Chromosome 4 of *A. thaliana* is the smallest of its five chromosomes and presents several remarkable features (Fig. 1). It has an acrocentric architecture with a long arm 14.6 Mb long and short arm about 8 Mb long tipped by the nucleolar organizer region (NOR). This region is about 3.6–4 Mb long and is constituted of almost homogeneous ribosomal DNA repeats (Haberer et al. 1996). The available short arm sequence starts in the last proximal copy of the rDNA repeat (Mayer et al. 1999; The *Arabidopsis* Information Resource, http://www.arabidopsis.org/). In some accessions, including Columbia (Col) but not Landsberg (Ler), the short arm has a heterochomatic region, called the "knob," identified cytologically (Fransz et al. 2000), primarily comprising transposable elements, in which a few genes are insulated (Mayer et al. 1999; Lippman et al. 2004). Moreover, an approximately 1.5-Mb-long region of the short arm, including the knob, is inverted between the two accessions, Col and Ler (Fransz et al. 2000).

We genotyped a population of 736 F2 plants resulting from a cross between Col and Ler (see Methods) with 71 SNPs (Supplemental Table 1) chosen from the Monsanto database (Jander et al. 2002) to be evenly spaced on the *Arabidopsis* chromosome 4. The average interval between two SNPs was 204 kb on the long arm (60 SNPs) and 239 kb on the short arm (11 SNPs).

Variation of CO rates across chromosome 4

After SNP genotyping, we analyzed the variation in CO rates in 702 plants (34 plants had missing data for more than 24 markers and were thus discarded). On average, we genotyped 666 plants (thus representing 1332 meioses because in an F2 plant each chromosome comes from an independent meiosis) per interval. We verified that there was no bias in the segregation of each marker. The cumulated genetic distance of the chromosome was estimated to be 83.9 cM, of which 69 cM corresponded to the long arm (Supplemental Table 2).

As the intervals were small, the genetic length of each interval can be simply calculated by dividing the number of recombinant chromosomes by the number of meioses analyzed. Genetic recombination varied greatly along the chromosome, from

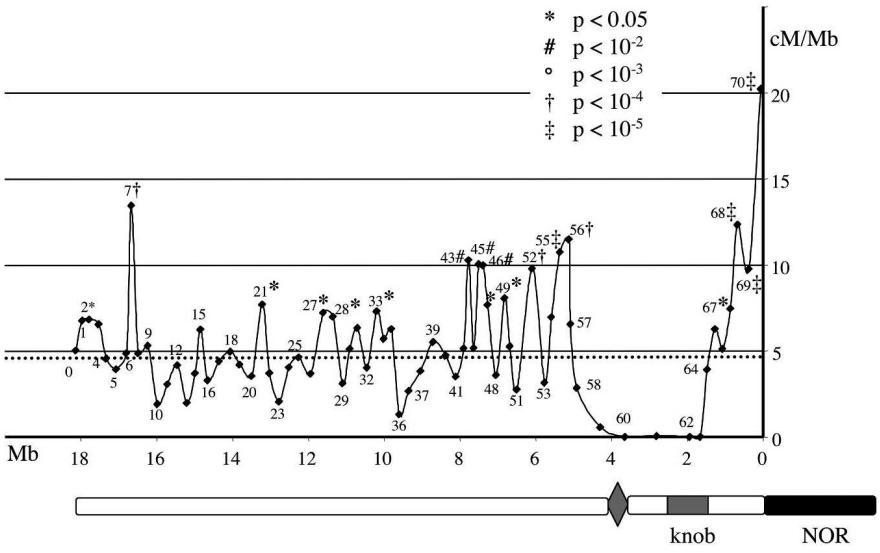


Figure 1. Variation of the CO rates on chromosome 4 of *A. thaliana*. The numbers refer to the intervals given in Supplemental Table 2. The dotted line represents the average CO rate on chromosome 4 (4.6 cM/Mb). A schematic representation of chromosome 4 of *A. thaliana* is aligned with the diagram. (Black box) NOR (nucleolar organizer region), (gray box) heterochromatic knob, (diamond-shaped box) centromere.

0 cM/Mb next to the centromere, to 20.2 cM/Mb next to the NOR (Supplemental Table 2; Fig. 1). The frequencies of COs in different intervals could not be directly compared because of both the variation in interval length and the number of analyzed chromosomes. Therefore, we developed a statistical approach to unambiguously identify intervals that were significantly either "colder" or "hotter" than the chromosome average. The approach is based on a simply binomial model of the number of COs in each interval, so that the "temperature" of an interval is determined by the probability that the number of COs in it exceeds the expected one, under the assumption that the recombination rate is constant along the chromosome. We implemented a statistical program (TETRA) to compute both the average number of COs per nucleotide, and the significance of the observed values from the binomial model (see Methods).

TETRA calculated an average of 4.6×10^{-8} COs/nucleotide, which is, on average, 1 cM for 217 kb for chromosome 4. Among the 70 intervals tested, TETRA identified 30 intervals with a significant deviation from the average rate of COs; 12 intervals had a significantly lower rate (cold) and 18 had a significantly higher rate (hot) ($P > 0.95$ and $P < 0.05$ for the cold and hot intervals, respectively; Supplemental Table 2). The hot intervals were not randomly distributed: four (intervals 67–70) were clustered on the short arm next to the NOR and eight (intervals 43–56) were clustered in a 3-Mb region on the long arm next to the centromere (Fig. 1). There was almost no genetic recombination in the centromeric and inverted region (intervals 58–63) and no clustering of the cold intervals was observed outside the centromeric region. In the middle of the long arm, there were alternate hot and cold intervals, although the "temperature" of most of these intervals was not significantly different from the chromosome average. In summary, the COs were unevenly distributed along chromosome 4 with alternating hot and mildly cold regions.

Correlation of CO rates with primary sequence features

We performed a principal component analysis to determine the most relevant genome features that correlated with the observed CO frequencies. The genes, pseudogenes, G+C content, and CpG log ratio, as well as repeated sequences, such as transposable elements (TEs) and single repeats (SSR), were carefully listed from both publicly available data and in-house computed analysis (see Methods). In each interval, we took the G+C content and the CpG ratio (see Methods) and calculated the density of each of the other features. We analyzed the whole chromosome, excluding the intervals 60–63 contained in the inverted region. On the first principal component axis, accounting for 46.4% of the variation, we found that gene, pseudogene, and TE densities contribute the most to the composition diversity of the intervals (Fig. 2). However, this axis shows that these features do not occur randomly along the chromosome, but follow two opposite gradients: The gene density is low in the pericentromeric and subtelomeric regions and high in the middle of chromosome arms, whereas the opposite is true for pseudogene and TE density. On the second principal component axis, adding 22.6% to the explained variation, the GC content and the CpG ratio appear to be more relevant to CO rates' variation. The intervals showing a significantly higher rate of COs tend to cluster regions of low G+C content where the CpG ratio is high. Conversely, the intervals with a low CO rate cluster in regions of high G+C content where the CpG ratio is low (Fig. 2).

A regression analysis carried out between the CO rate and the G+C content or CpG ratio confirmed these trends with R^2 of 0.18 ($P = 3 \times 10^{-4}$) for G+C content and an R^2 of 0.20 ($P = 1.3 \times 10^{-4}$) for the CpG ratio. The regression was stronger when analyzing only the long arm of the chromosome, with $R^2 = 0.36$ ($P = 4 \times 10^{-7}$) for G+C content and $R^2 = 0.22$ ($P = 1.7 \times 10^{-4}$) for the CpG ratio. Of the other regressions tested (gene density, pseudogenes, etc.), only the SSR density had a significant correlation with CO rates ($R^2 = 0.13$; $P = 3 \times 10^{-3}$). Therefore, unlike the results obtained in several other eukaryotes, in which a high CO rate tends to correlate with a high G+C content, we suggest that on chromosome 4 of *A. thaliana* a high

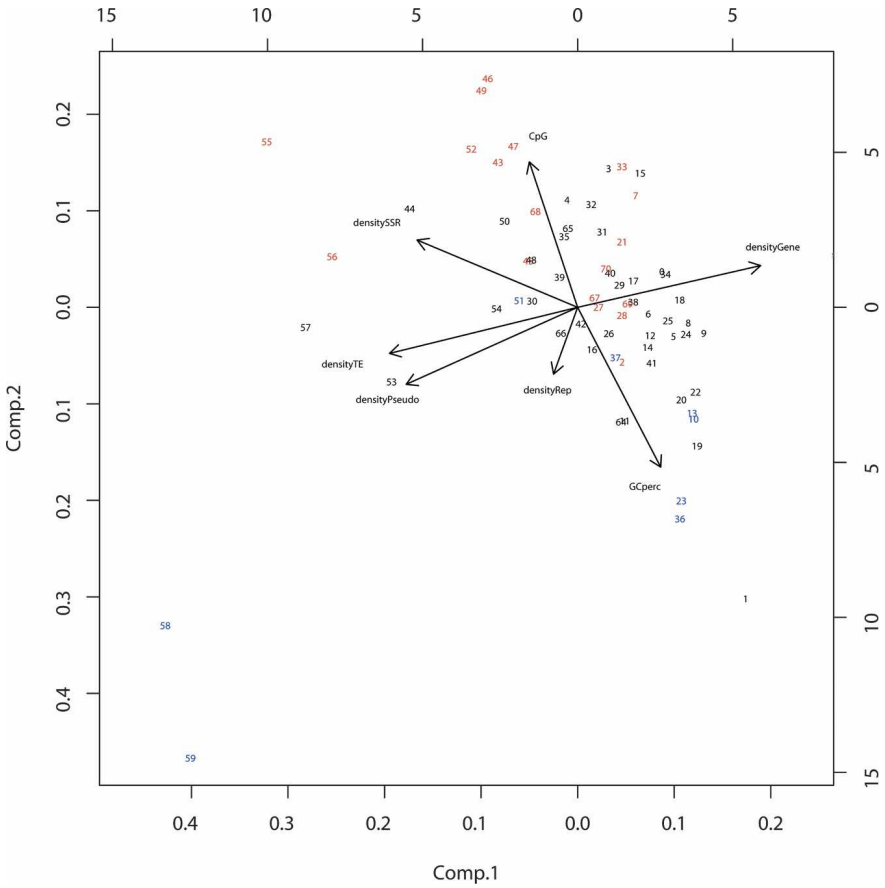


Figure 2. Principal component analysis of chromosome 4 of *A. thaliana*. Numbers refer to the intervals given in Figure 1 and Supplemental Table 2. Hot intervals are indicated in red; cold intervals are indicated in blue.

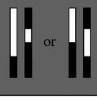
COs / plant	COs / chromatid: non ambiguous (number of plants)	Examples	COs / chromatid: ambiguous (number of plants)	Examples
0	0+0 (100)		none	
1	0+1 (223)		none	
2	1+1 (97)		2: 1+1 or 0+2 (123)	
2	0+2 (38)			
3	1+2 (57)		3: 0+3 or 2+1 (41)	
>3	none		> 3 (23)	

Figure 3. Number of CO events per chromatid deduced from the genotype of an F2 plant. When an F2 plant displays two COs, either the extremities of both chromosomes are homozygous, and the COs are on one chromatid or both, or the extremities are heterozygous and there is ambiguity between one event on each chromatid or two events on the same chromatid. A similar approach has been used to analyse F2 plants that displays three COs. When an F2 plant displays more than three COs, the recombination history cannot be inferred from the genotype. In the columns "COs/chromatid," the numbers in parentheses represent the number of plants in each category. (Light gray box) plants used in the interference analysis. (Dark gray box) plants containing a chromatid with two COs but that could not be used in the interference analysis.

CO rate correlates with a low G+C content. The CpG ratio and SSR density also weakly correlate with CO rates.

Interference on chromosome 4

We obtained 1171 COs for 1404 analyzed meioses. This corresponded to an average of 0.8 events per chromatid and per meiosis, corresponding to 1.6 COs per pair of homologous chromosomes (bivalents) per meiosis. There were, on average, 1.3 events on the long arm and 0.3 events on the short arm. However, if we take into account the 1.5 Mb that are inverted between the two parental lines, and therefore "forbidden" from forming and/or recovering COs, the ratio of COs per megabase on the short arm was double that of the long arm (0.18 vs. 0.09).

For 515 pairs of chromosomes, we were able to determine unequivocally the number of exchanges that each chromatid had undergone (0, 1, or 2 COs) during meiosis (Fig. 3). For 123 pairs of chromosomes harboring two exchanges, we could not unambiguously attribute the recombination events to one or the other chromatid. We reassigned them either to the "1 + 1" or the "2 + 0" class (see Fig. 3) on the basis of the prorata between the sizes of these latter classes in the nonambiguous class with two exchanges. The 41 pairs that exhibit three exchanges that could not be credited to one or the other chromatid were considered to fall in the "2 + 1" class (that is, we assumed "3 + 0" pairs to be very rare). For the remaining pairs (23), which display four or more CO, we could not attribute CO unambiguously to parental chromosomes, so we discarded them. As expected, one exchange event was the most common occurrence (692 chromatids). Furthermore, we compared the observed distribution of the number of COs to what is expected under a Poisson distribution (Supplemental Table 3), Test of χ^2 goodness-of-fit shows that the null Poisson hypothesis can be strongly rejected ($\chi^2 = 121.8$, $P < 5 \times 10^{-4}$). Hence, we can conclude that multiple COs do not occur on chromosome 4 independently one from each other.

For each of the 38 plants having two precisely located COs on the same chromatid (Fig. 3, light gray box), we calculated the genetic distance between the two COs. The distance varied from 1.17 to 62.8 cM with a mean distance of 44.1 cM. The mean expected value for randomly distributed double COs was one-third of the chromosome, being 27.9 cM (see Methods). We then classified the 38 plants into four groups: group 1, with events separated by less than 25% of the chromosome (0–21 cM); group 2, with events separated by more than 25% but less than 50% of the chromosome (21–42 cM); group 3, with events separated by more than 50% but less than 75% of the chromosome (42–63 cM); and group 4, with events separated by more than 75% of the chromosome (63–83.9 cM) (Fig. 4). We compared the observed distribution with the expected distribution if COs were located independently of each other. We found a very strong probability ($\chi^2 = 27.9$, $P < 5 \times 10^{-3}$) that double COs were not located independently of each other. The same analysis on only the long arm also showed that the observed distribution and the observed mean distance (36 cM) were very different from the theoretical values (23 cM; data not shown). We then looked at the effect of the centromere on interference. For the 12 chromosomes having one CO on the short arm and the other on the long arm, the mean distance was 59.6 cM, that is, 70% of the genetic length of the chromosome, while the mean distance between two COs occurring on the long arm represents 52% of the genetic length of the long arm. These results confirm that CO location on chromosome 4 is affected by interference and that the centromere is not a barrier to interference.

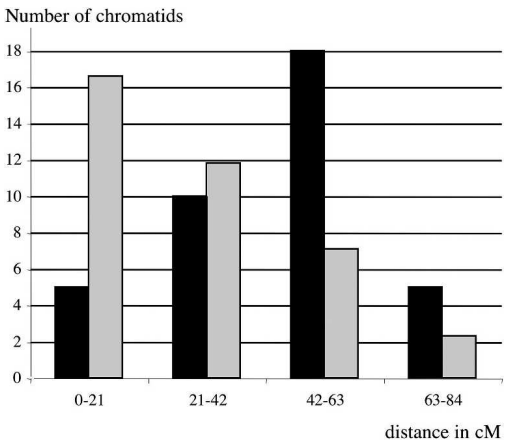


Figure 4. Distribution of the distances in centiMorgans between double COs. (Histogram in black) observed distribution of double COs in our F2 (see text); (histogram in gray) theoretical distribution of double COs if the position of one CO is independent of the second (Methods).

Evidence for the existence of hot spots of recombination

We further investigated several of the 14 intervals having the highest CO rates together with one interval with a slightly above average CO rate and one cold interval. For each interval, we genotyped the corresponding recombinant plants using a set of SNP or indel markers, giving precise locations of the exchange points. We divided the hottest interval (interval 70; Fig. 1) into 15 parts to map the COs at a precision of a few kilobases (Fig. 5A). We found a clearly nonhomogeneous distribution of exchange events. Two very small fragments (3.4 and 3.2 kb) 20 kb apart exhibited a very high rate of COs (>85 cM/Mb), being 15 times higher than the chromosome average (4.6 cM/Mb) and four times higher than the interval average (20.2 cM/Mb). We found that two other fragments in interval 70 had moderately high rates of genetic recombination (40 and 55 cM/Mb, 8 to 10 times the chromosome average). We also analyzed another hot interval (interval 21, Fig. 5B) in the middle of the long arm (Fig. 1). We found one DNA fragment displaying a large increase of genetic recombination in this interval. The recombination rates in the remainder of this interval were mostly lower than the chromosome average. We also observed the same type of "spotty" CO distribution in the other hot intervals that we investigated (7, 55, 56 and 68, 69; data not shown).

We then analyzed interval 57 (Fig. 1), which did not appear to have a significantly high rate of genetic recombination when analyzed by TETRA (6.6 cM/Mb; $P = 0.08$). We found one DNA fragment of 12 kb displaying a high rate of genetic recombination (40 cM/Mb) whereas the remainder of the interval displayed CO rates below the chromosome average (Fig. 5C). Interval 37 was found to be significantly cold when analyzed by TETRA (2.7 cM/Mb; $P > 0.98$). We performed the same kind of analysis as for

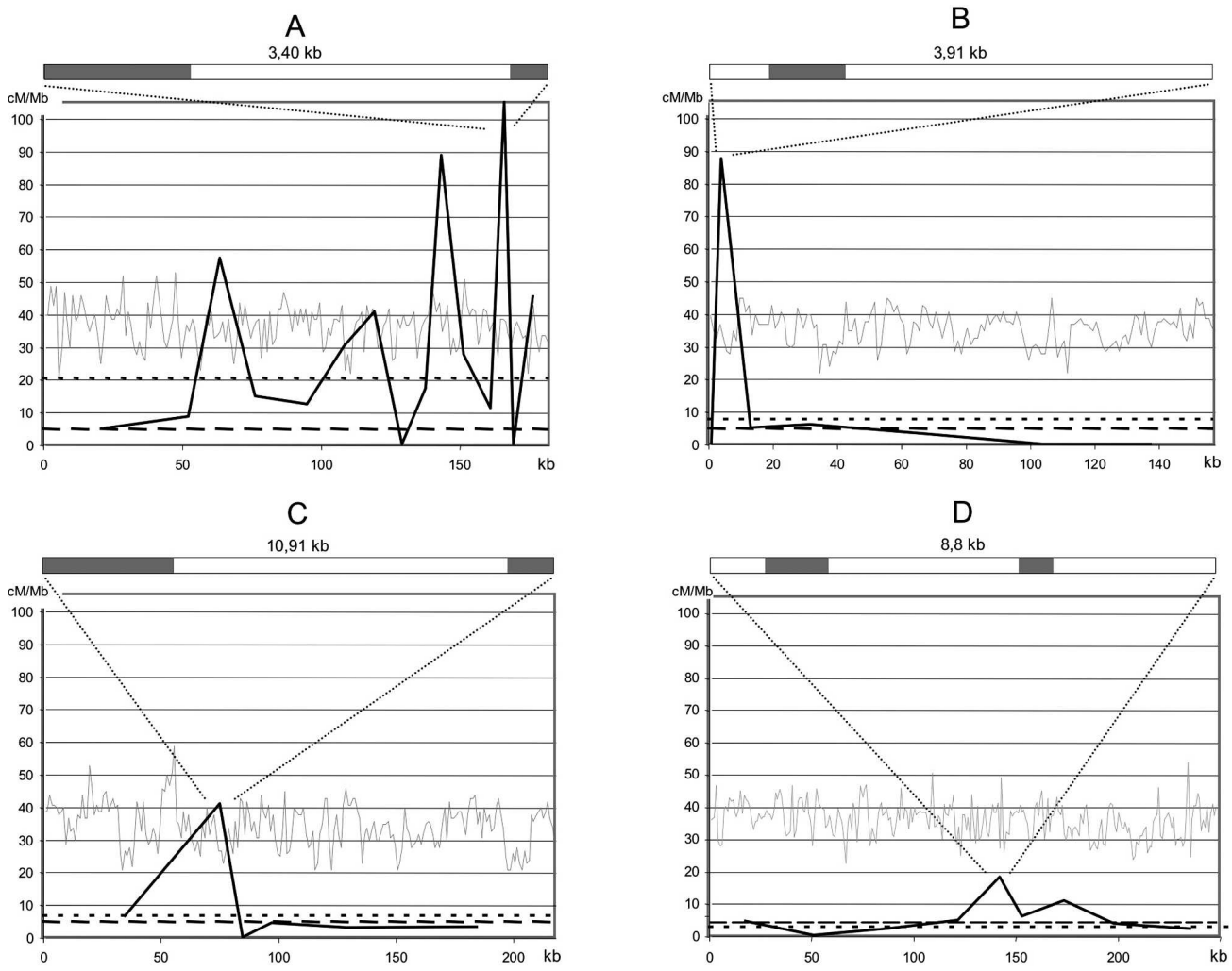


Figure 5. Fine-scale analysis of the distribution of CO breakpoints in four intervals. (*A*) interval 70: 44.5kb-3; 16.4kb-2; 6.2kb-5; 19.5kb-4; 17.5kb-3; 9.4kb-4; 12.3kb-7; 8.1kb-0; 8.3kb-2; 3.2kb-4; 13.0kb-5; 6.4kb-1; 3.4kb-5; 3.3kb-0; 11.0kb-7. (*B*) interval 21: 2.5kb-0; 3.9kb-5; 13.8kb-1; 23.6kb-2; 45.7kb-2; 29.2kb-0; 38.6kb-0. (*C*) interval 57: 69.5kb-6; 12.2kb-7; 7.1kb-0; 17.4kb-1; 46.5kb-2; 65.0kb-3. (*D*) interval 37: 35.2kb-2; 32.7kb-0; 38.1kb-1; 8.8kb-2; 12.6kb-1; 28.3kb-1; 20.2kb-1; 49.2kb-2; 38.7kb-1. For each interval the size of each DNA fragment is given and the number of recombinant plants. (Light gray line) G+C content calculated in 1-kb windows. (Black line) CO rates in centiMorgans per megabase. (Small dotted line) interval COs rate average. (Large dotted line) chromosome 4 COs rate average. Above each major peak is the gene organization of the fragment. (Gray box) gene.

the other intervals. We found a dispatch of the 12 CO exchanges in 9 of the 10 fragments studied (Fig. 5D) with a maximum of two events in an 8.8-kb fragment. This small fragment seems to exhibit a slightly higher CO rate than the genome average (Fig. 5D). However, more plants would be needed to confirm this difference. For the four regions analyzed, hot spots did not seem to correlate with G+C content or gene organization (Fig. 5A–D).

Discussion

We obtained a very detailed genetic map of chromosome 4 of *A. thaliana* by genotyping a series of 71 SNP markers on 702 F2 plants issued from an F1 Col/Ler hybrid. The total size of the genetic map was estimated at 83.9 cM, which is consistent with other maps obtained from crosses of the same accessions: the classical map (76 cM; Meinke et al. 1998), the RFLP map (74.4 cM; Schmidt et al. 1995; Liu et al. 1996), tetrad analysis using the *quartet* mutation (85 cM; Copenhaver et al. 1998; Lam et al. 2005), and first versions of the RIL genetic map (76 cM; Lister and Dean 1993).

We found that, on average, a chromosome 4 bivalent undergoes 1.6 crossovers per meiosis. Copenhaver et al. found an average of 1.5 COs on chromosome 4 in male meiosis in a Col/Ler cross (Copenhaver et al. 1998; Lam et al. 2005). Meiotic recombination has also been assessed using cytology by recording the numbers and locations of chiasmata on metaphase I bivalents in pollen mother cells of several accessions, including Col and Ler (Sanchez-Moran et al. 2002). Both the genetic and cytological methods gave consistent results, with the mean chiasma frequency being 1.6 for chromosome 4. Therefore, CO frequency on chromosome 4 during meiosis of a Col/Ler F1 hybrid is not greatly different from that in the parents.

In most eukaryotes, "positive interference" (i.e., the probability of COs occurring next to each other is lower than expected) affects the distribution of multiple COs on a single chromosome (see Zickler and Kleckner 1999). However, not all the COs seem to interfere, and recent data suggest two pathways for crossovers in *S. cerevisiae*, in humans, and in *A. thaliana*: one pathway being sensitive to interference (class I) and the other insensitive (class II) (Copenhaver et al. 2002; Housworth and Stahl 2003; Higgins et al. 2004; Hollingsworth and Brill 2004; Stahl et al. 2004; Lam et al. 2005; Mercier et al. 2005). We show also that COs are subjected to interference on chromosome 4. Double COs on the same chromatid are significantly further than one-third of the chromosome length apart, contrary to what is expected for randomly distributed COs. In addition, our results suggest that interference is insensitive to centromere as previously proposed by Colombo and Jones (1997) and that the centromere may increase the strength of interference on chromosome 4. However, there is not complete interference, as we observed double COs only a few centiMorgans apart. We could assume, as suggested by previous studies (Copenhaver et al. 2002; Lam et al. 2005) that these close double COs are insensitive to interference and the distant double COs are sensitive to interference. In yeast, the level of interference has been shown to depend on the size of the chromosome with the short chromosomes harboring less interference (Kabback et al. 1999). However, the disparity in size of the chromosomes is less pronounced in *Arabidopsis*, with the shortest chromosome being more than two-thirds of the size of the longest chromosome, while in yeast there is a fourfold difference in size. Moreover, Lam et al. (2005) recently provided evidences that in *Arabidopsis* NOR-bearing chromosomes (i.e., chromosomes 2 and 4) exhibit more interference than the others and suggested that the NOR region itself rather than the size of the chromosome could influence interference. Further analyses are needed to determine whether interference varies along the chromosome, as recently suggested in a study on rice (Esch 2005).

Numerous studies have attempted to understand the factors responsible for genetic recombination variations and to identify primary sequence features that may correlate with this variability. In many sexual organisms, such as mammals, birds, yeast, drosophila, and nematodes, positive correlations between the CO rates and G+C content have been observed at the scale of several hundred of kilobases (Hurst et al. 1999; Gerton et al. 2000; Fullerton et al. 2001; Marais et al. 2001; Takano-Shimizu 2001; Yu et al. 2001; Birdsell 2002; Kong et al. 2002; Jensen-Seaman et al. 2004). Gerton et al. (2000) suggested that regions of high G+C content stimulate recombination. Alternatively, several recent studies have proposed that high levels of recombination may create regions with high G+C content, probably through a biased gene conversion (BGC) toward G+C. In other words, meiotic recombination modifies the base composition through the average density of recombination hot spots (see below; Galtier et al. 2001; Birdsell 2002; Montoya-Burgos et al. 2003; Meunier and Duret 2004). However, in humans, rats, and mice, when CpG ratio is included in a multiple regression model, the correlation with the G+C content becomes negative (Kong et al. 2002; Jensen-Seaman et al. 2004).

In contrast, we found that regions of low G+C content and high CpG ratio on chromosome 4 of *A. thaliana* tend to have higher rates of genetic recombination. Therefore, the BGC hypothesis suggested to explain the correlation found in other eukaryotes may not apply in *Arabidopis*. However, homologs of genes believed to participate in G+C-biased mismatch repair in other organisms exist in the genome of *Arabidopsis* (Birdsell 2002). It is also possible that in *Arabidopsis* BGC cannot affect the nucleotide content due to the high level of inbreeding of the plant that does not favor the formation of heteroduplex DNAs. However, this would explain an absence of correlation but not a negative correlation. In contrast to our results, a study recently reported no correlation between the G+C content and CO rates in *Arabidopsis* (Marais et al. 2004). We suggest that our observation is due to the higher precision of our recombination map because we studied 702 plants compared to the 101 RILs used in the study of Marais et al. (2004). Therefore, our study in *Arabidopsis* questions the assumptions made for G+C correlation and so, the problem of causation remains an open query. Data from more species are needed and may reveal a species-specific lineage in the evolution of recombination.

A fine-scale analysis of several intervals showed peaks in crossover activity. For example, in the hottest interval (interval 70; Fig. 1), CO breakpoints are found in 12 of the 14 fragments tested, even though there is clustering in two small regions 20 kb apart (Fig. 5A). In other intervals, including one not having a significantly high CO rate, one small DNA fragment accounts for most of the genetic recombination of the interval. In the genome of *A. thaliana,* this punctuate distribution of CO activity strongly suggests recombination hot spots where recombination events group around an initiation site. In plants, several hot spots of CO activity have been described. The 140-kb *a1-sh2* region in maize has peaks of CO activity (three to six times the genome average) in three small intervals (1.7–3.4 kb) (Yao et al. 2002). Other loci in maize or in rice, such as *bronze* or *waxy*, show some properties

of hot spots (Dooner and Martinez-Ferez 1997; Okagaki and Weil 1997; Inukai et al. 2000). For the *bronze* locus, unlike for yeast and mammals, it has been suggested that recombination is initiated uniformly along the gene and not at a preferential site. Therefore, although existence of hot spots seems to be the rule rather than the exception in plants and other higher eukaryotes, there may be some differences. However, all studies on higher eukaryotes looked at no more than one or two intervals that were often selected for a phenotype associated with the recombination event. Therefore, it is difficult to determine whether the observed hot spot patterns in these intervals can be applied to the whole genome. Here, we show that a punctuate distribution of hot spots is a general feature of the chromosome that is not restricted to significantly recombinogenic regions. Our results also strongly suggest that recombination is initiated at preferential sites all along the chromosome. However, both the intensity and the density of the recombination sites influence the variation of recombination, as hot regions contain one or several very hot spots whereas a mildly warm interval would contain only one mild hot spot. Cold regions may contain few spots with a higher rate of recombination than the genome average, but it remains to be demonstrated. A similar result has recently been obtained in the human genome, where strong hot spots have been detected in narrow regions of strong LD and weak hot spots in regions of strong marker association (Jeffreys et al. 2005). We have identified more than 10 small DNA fragments that may behave as hot spots on chromosome 4 of *A. thaliana*. Further experiments are needed to confirm the strength and the precise location of the initiation site of these hot spots. Their fine characterization and analysis in other genetic backgrounds is needed to determine the factors that govern their activity and distribution.

Methods

F2 recombinant population construction, genomic DNA extraction

The two *Arabidopsis* accessions, Columbia and Landsberg *erecta*, were crossed to obtain an F1 hybrid. Self-fertilization from a single F1 was carried out to obtain F2 seeds. Seeds were grown in soil in long-day conditions in the greenhouse. At the rosette stage, the whole material of 736 F2 plants together with plant material from the two parental accessions was collected. DNA was extracted as described (Loudet et al. 2002).

Selection of SNPs

Most of the SNPs were chosen from the Monsanto database (Jander et al. 2002). When convenient SNPs were not found in the database, DNA fragments were amplified at the desired position on the genomic DNA of the two parental accessions and sequenced to identify a SNP suitable for genotyping. A list of the SNPs used in this study is given in Supplemental Table 1. A couple of primers were designed for each SNP to obtain a PCR fragment containing the predicted SNP. A list of the PCR primers used in this study is given in Supplemental Table 1. The PCRs were carried out on the parental accession DNAs using standard conditions: 94°C 4 min, (94°C 45 sec, 52°C 45 sec, 72°C 1 min) × 35 cycles, 72°C, with Eurobio 1× reaction buffer and Taq polymerase. PCR fragments were sequenced (Genome Express) to check the presence and position of the SNPs.

SNP genotyping

At each of the SNP sites, DNA extracted from the F2 plants was genotyped either by matrix-assisted laser desorption/ionization time-of-flight (MALDI-TOF) mass spectrometry, as described by Sauer et al. (2000) or by fluorescence based techniques: the Amplifluor technology (Serological Corporation) or the TaqMan technology (Applied Biosystems). For a number of SNPs, the results obtained with one technique (usually mass spectrometry) were confirmed with one of the two other methods. The techniques used for each SNP are given in Supplemental Table 1.

Statistical analysis of CO rates: TETRA

We define P as the probability of having a CO in a specific position of the chromosome, and assume that $P \ll 1$. The probability P_i of observing one CO in the ith interval on a chromosome is approximated as PL_i, where L_i stands for the length of the ith interval. If the number of informative chromosomes for the interval i (i.e., the number of chromosomes for which both SNPs delimiting the interval are available) is written as V_i, then the number, N_i, of COs in the ith interval is distributed according to a binomial $B(V_i,PL_i)$ under the null hypothesis that the COs rate is constant along the chromosome. More explicitly, we have for all

$$k\text{: } P(N_i = k) = \binom{Vi}{k}(P \times L_i)^k\,(1 - P \times L_i)^{Vi-k}\,.$$

According to the observed values, n_i, of the number of COs in the different intervals, TETRA computes the average CO rate, P, along the whole chromosome. It then computes the P-value, T_i, of the observed number of COs under the above binomial model, that is:

$$P(N_i \geq n_i) = \sum_{k=ni}^{Vi} \binom{Vi}{k}(P \times L_i)^k(1 - P \times L_i)^{Vi-k}.$$

This P-value can be interpreted as the probability that the number of COs in the ith interval exceeds its observed value under the model of homogeneous CO rate along the chromosome.

Statistical analysis of COs interference

We derive the probability distribution function of the distribution of distances between the two COs by assuming that the locations of the two COs are independently uniformly distributed random variables. L is the length of the chromosome, and x and y are locations of the two COs. x and y are uniformly distributed in $[0, L]$. The distance $r = (x - y)$. The distribution of x and y is symmetric under the exchange of x and y; we can condition on $x > y$. The probability distribution function $P(r)$ is proportional to the length of the segment, in the x, y plane, between the points of coordinates $(r, 0)$ and $(L, L - r)$, which is itself proportional to $L - r$. Imposing the normalization of the probability distribution function, we obtain $P(r) = 2(1 - r/L)$. The expectation value of r is thus $\int_0^L P(r)r\,dr = L/3$. We can deduce the probabilities P_1, P_2, P_3, P_4 of r being in each of the four bins $[0, L/4]$, $[L/4, L/2]$, $[L/2, 3L/4]$, $[3L/4, L]$. For instance $P_1 = 2\int_0^{1/4}(1 - x)dx = 7/16$. Similarly, we find $P_2 = 5/16$, $P_3 = 3/16$, $P_4 = 1/16$. These probabilities are used in the analysis of Figure 4.

Correlation studies

The *A. thaliana* genomic sequence and its annotation were downloaded from the TIGR Web site (http://ftp.tigr.org/pub/data/a_thaliana/ath1/). Gene and pseudogene annotations have been

extracted from TIGR-XML files from release 5 of the genome annotation. Transposable elements have been re-annotated using the RMBLR procedure from the TE annotation pipeline described by Quesneville et al. (2005). The TE reference set used is derived from the *A. thaliana* RepeatMasker repeat library (March 6, 2004). The same TE family consecutive TE fragments (on both the genome and the reference TE) have been automatically joined if separated by a sequence composed of more than 80% of other TE insertions (in this case we have a nested TE). Otherwise they are joined if a gap of 5000 nucleotides or a region of mismatches 500 nucleotides long separate them.

Single repeats (SSR) were found using the Tandem Repeat Finder program (Benson 1999), and repeats by a BLASTN all-by-all using BLASTER and GROUPER (Quesneville et al. 2005) without using any simple link clustering coverage constraint. G+C content and CpG were counted with in-house python scripts. CpG ratios were computed by taking the $\log_{10}$ of the C+G dinucleotide frequency divided by the product of G and C frequencies. All data and analysis results were stored in a MySQL database and retrieved by SQL queries.

Statistical analyses were carried out using the R software environment (http://cran.r-project.org)

Detection of hot spots

We halved each interval by choosing a convenient SNP (or indel) either from the Monsanto database or by DNA sequencing (see above). We then sequenced the DNA fragment containing the SNP in the recombinant plants and finally distributed the plants according to their genotype within one or the other half interval. This was iteratively repeated until the location of CO breakpoints was obtained within a few kilobases. A list of the SNPs, indels, and corresponding primers is given in Supplemental Table 1. Genomic DNA from plants was amplified by PCR in standard conditions (see above) with an annealing temperature adapted for each set of primers.

Acknowledgments

We thank Mathilde Grelon, Raphaël Mercier, Eric Jenczewski, and Valérie Borde for critical reading of the manuscript and Sylvie Jolivet for technical assistance. All the members of the "Méiose et Recombinaison" group provided helpful comments and participated in stimulating discussions. Marc Mézard kindly provided the statistical analysis of interference. This work was supported by grants from the Institut National de la Recherche Agronomique (to C.M.) and the European Union (Epigenome Network of Excellence to V.C.).

References

Anderson, L.K. and Stack, S.M. 2002. Meiotic recombination in plants. *Curr. Genomics* **3:** 507–525.

The *Arabidopsis* Genome Initiative 2000. Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature* **408:** 796–815.

Baudat, F. and Nicolas, A. 1997. Clustering of meiotic double-strand breaks on yeast chromosome III. *Proc. Natl. Acad. Sci.* **94:** 5213–5218.

Benson, G. 1999. Tandem repeats finder: A program to analyze DNA sequences. *Nucleic Acids Res.* **27:** 573–580.

Birdsell, J.A. 2002. Integrating genomics, bioinformatics, and classical genetics to study the effects of recombination on genome evolution. *Mol. Biol. Evol.* **19:** 1181–1197.

Borde, V., Lin, W., Novikov, E., Petrini, J.H., Lichten, M., and Nicolas, A. 2004. Association of Mre11p with double-strand break sites during yeast meiosis. *Mol. Cell* **13:** 389–401.

Carpenter, A.T.C. 1988. Thoughts on recombination nodules, meotic recombination, and chiasmata. In *Genetic recombination* (eds. E.R. Kucherlapati and G.R. Smith), pp. 529–548. American Society for Microbiology, Washington, DC.

Colombo, P.C. and Jones, G.H. 1997. Chiasma interference is blind to centromeres. *Heredity* **79:** 214–227.

Copenhaver, G.P., Browne, W.E., and Preuss, D. 1998. Assaying genome-wide recombination and centromere functions with *Arabidopsis* tetrads. *Proc. Natl. Acad. Sci.* **95:** 247–252.

Copenhaver, G.P., Housworth, E.A., and Stahl, F.W. 2002. Crossover interference in *Arabidopsis*. *Genetics* **160:** 1631–1639.

de Massy, B. 2003. Distribution of meiotic recombination sites. *Trends Genet.* **19:** 514–522.

Dooner, H.K. and Martinez-Ferez, I.M. 1997. Recombination occurs uniformly within the bronze gene, a meiotic recombination hotspot in the maize genome. *Plant Cell* **9:** 1633–1646.

Esch, E. 2005. Estimation of gametic frequencies from F2 populations using the EM algorithm and its application in the analysis of crossover interference in rice. *Theor. Appl. Genet.* **111:** 100–109.

Fransz, P.F., Armstrong, S., de Jong, J.H., Parnell, L.D., van Drunen, C., Dean, C., Zabel, P., Bisseling, T., and Jones, G.H. 2000. Integrated cytogenetic map of chromosome arm 4S of *A. thaliana*: Structural organization of heterochromatic knob and centromere region. *Cell* **100:** 367–376.

Fullerton, S.M., Carvalho, A.B., and Clark, A.G. 2001. Local rates of recombination are positively correlated with GC content in the human genome. *Mol. Biol. Evol.* **18:** 1139–1142.

Galtier, N., Piganeau, G., Mouchiroud, D., and Duret, L. 2001. GC-content evolution in mammalian genomes: The biased gene conversion hypothesis. *Genetics* **159:** 907–911.

Gerton, J.L., DeRisi, J., Shroff, R., Lichten, M., Brown, P.O., and Petes, T.D. 2000. Global mapping of meiotic recombination hotspots and coldspots in the yeast *Saccharomyces cerevisiae*. *Proc. Natl. Acad. Sci.* **97:** 11383–11390.

Gut, I.G. 2001. Automation in genotyping of single nucleotide polymorphisms. *Hum. Mutat.* **17:** 475–492.

Haberer, G., Fischer, T.C., and Torres-Ruiz. R.A. 1996. Mapping of the nucleolus organizer region on chromosome 4 in *Arabidopsis thaliana*. *Mol. Gen. Genet.* **250:** 123–128.

Higgins, J.D., Armstrong, S.J., Franklin, F.C., and Jones, G.H. 2004. The *Arabidopsis* MutS homolog AtMSH4 functions at an early step in recombination: Evidence for two classes of recombination in *Arabidopsis*. *Genes & Dev.* **18:** 2557–2570.

Hollingsworth, N.M. and Brill, S.J. 2004. The Mus81 solution to resolution: Generating meiotic crossovers without Holliday junctions. *Genes & Dev.* **18:** 117–125.

Housworth, E.A. and Stahl, F.W. 2003. Crossover interference in humans. *Am. J. Hum. Genet.* **73:** 188–197.

Hurst, L.D., Brunton, C.F., and Smith, N.G. 1999. Small introns tend to occur in GC-rich regions in some but not all vertebrates. *Trends Genet.* **15:** 437–439.

Inukai, T., Sako, A., Hirano, H.Y., and Sano, Y. 2000. Analysis of intragenic recombination at wx in rice: Correlation between the molecular and genetic maps within the locus. *Genome* **43:** 589–596.

Jander, G., Norris, S.R., Rounsley, S.D., Bush, D.F., Levin, I.M., and Last, R.L. 2002. *Arabidopsis* map-based cloning in the post-genome era. *Plant Physiol.* **129:** 440–450.

Jeffreys, A.J., Neumann, R., Panayi, M., Myers, S., and Donnelly, P. 2005. Human recombination hot spots hidden in regions of strong marker association. *Nat. Genet.* **37:** 601–606.

Jensen-Seaman, M.I., Furey, T.S., Payseur, B.A., Lu, Y., Roskin, K.M., Chen, C.F., Thomas, M.A., Haussler, D., and Jacob, H.J. 2004. Comparative recombination rates in the rat, mouse, and human genomes. *Genome Res.* **14:** 528–538.

Jones, G.H. 1984. The control of chiasma distribution. *Symp. Soc. Exp. Biol.* **38:** 293–320.

Jones, G.H. 1987. Chiasmata. In *Meiosis* (ed. P.B. Moens), pp. 213–244. Academic Press, London.

Kaback, D.B., Barber, D., Mahon, J., Lamb, J., and You J. 1999. Chromosome size-dependent control of meiotic reciprocal recombination in *Saccharomyces cerevisiae*: The role of crossover interference. *Genetics* **152:** 1475–1486.

Kauppi, L., Jeffreys, A.J., and Keeney, S. 2004. Where the crossovers are: Recombination distributions in mammals. *Nat. Rev. Genet.* **5:** 413–424.

Keeney, S. 2001. Mechanism and control of meiotic recombination initiation. *Curr. Top. Dev. Biol.* **52:** 1–53.

Kong, A., Gudbjartsson, D.F., Sainz, J., Jonsdottir, G.M., Gudjonsson, S.A., Richardsson, B., Sigurdardottir, S., Barnard, J., Hallbeck, B., Masson, G., et al. 2002. A high-resolution recombination map of the human genome. *Nat. Genet.* **31:** 241–247.

Kwok, P.Y. 2001. Methods for genotyping single nucleotide polymorphisms. *Annu. Rev. Genomics Hum. Genet.* **2:** 235–258.

Lam, S., Horn, S.R., Radford, S.J., Housworth, E.A., Stahl, F.W., and Copenhaver, G.P. 2005. Crossover interference on NOR-bearing chromosomes in *Arabidopsis*. *Genetics* **170:** 807–812.
Lenormand, T. and Dutheil, J. 2005. Recombination difference between sexes: A role for haploid selection. *PLoS Biol.* **3:** e63.
Lippman, Z., Gendrel, A.V., Black, M., Vaughn, M.W., Dedhia, N., McCombie, W.R., Lavine, K., Mittal, V., May, B., Kasschau, K.D., et al. 2004. Role of transposable elements in heterochromatin and epigenetic control. *Nature* **430:** 471–476.
Lister, C. and Dean, C. 1993. Recombinant inbred lines for mapping RFLP and phenotypic markers in *Arabidopsis thaliana*. *Plant J.* **4:** 745–750.
Liu, Y.G., Mitsukawa, N., Lister, C., Dean, C., and Whittier, R.F. 1996. Isolation and mapping of a new set of 129 RFLP markers in *Arabidopsis thaliana* using recombinant inbred lines. *Plant J.* **10:** 733–736.
Loudet, O., Chaillou, S., Camilleri, C., Bouchez, D., and Daniel-Vedele, F. 2002. Bay-0 x Shahdara recombinant inbred line population: A powerful tool for the genetic dissection of complex traits in *Arabidopsis*. *Theor. Appl. Genet.* **104:** 1173–1184.
Lynn, A., Kashuk, C., Petersen, M.B., Bailey, J.A., Cox, D.R., Antonarakis, S.E., and Chakravarti, A. 2000. Patterns of meiotic recombination on the long arm of human chromosome 21. *Genome Res* **10:** 1319–1332.
Lynn, A., Koehler, K.E., Judis, L., Chan, E.R., Cherry, J.P., Schwartz, S., Seftel, A., Hunt, P.A., and Hassold, T.J. 2002. Covariation of synaptonemal complex length and mammalian meiotic exchange rates. *Science* **296:** 2222–2225.
Lynn, A., Ashley, T., and Hassold, T. 2004. Variation in human meiotic recombination. *Annu. Rev. Genomics Hum. Genet.* **5:** 317–349.
Marais, G., Mouchiroud, D., and Duret, L. 2001. Does recombination improve selection on codon usage? Lessons from nematode and fly complete genomes. *Proc. Natl. Acad. Sci.* **98:** 5688–5692.
Marais, G., Charlesworth, B., and Wright, S.I. 2004. Recombination and base composition: The case of the highly self-fertilizing plant *Arabidopsis thaliana*. *Genome Biol.* **5:** R45.
Mayer, K. Schuller, C., Wambutt, R., Murphy, G., Volckaert, G., Pohl, T., Dusterhoft, A., Stiekema, W., Entian, K.D., Terryn, N., et al. 1999. Sequence and analysis of chromosome 4 of the plant *Arabidopsis thaliana*. *Nature* **402:** 769–777.
Meinke, D.W., Cherry, J.M., Dean, C., Rounsley, S.D., and Koornneef, M. 1998. *Arabidopsis thaliana*: A model plant for genome analysis. *Science* **282:** 662, 679–682.
Mercier, R., Jolivet, S., Vezon, D., Huppe, E., Chelysheva, L., Giovanni, M., Nogue, F., Doutriaux, M.P., Horlow, C., Grelon, M., et al. 2005. Two meiotic crossover classes cohabit in *Arabidopsis*: One is dependent on MER3, whereas the other one is not. *Curr. Biol.* **15:** 692–701.
Meunier, J. and Duret, L. 2004. Recombination drives the evolution of GC-content in the human genome. *Mol. Biol. Evol.* **21:** 984–990.
Montoya-Burgos, J.I., Boursot, P., and Galtier, N. 2003. Recombination explains isochores in mammalian genomes. *Trends Genet.* **19:** 128–130.
Nachman, M.W. 2002. Variation in recombination rate across the genome: Evidence and implications. *Curr. Opin. Genet. Dev.* **12:** 657–663.
Okagaki, R.J. and Weil, C.F. 1997. Analysis of recombination sites within the maize waxy locus. *Genetics* **147:** 815–821.
Petes, T.D. and Merker, J.D. 2002. Context dependence of meiotic recombination hotspots in yeast: The relationship between recombination activity of a reporter construct and base composition. *Genetics* **162:** 2049–2052.
Quesneville, H., Bergman, C.M., Andrieu, O., Autard, D., Nouaud, D., Ashburner, M., and Anxolabehere, D. 2005. Combined evidence annotation of transposable elements in genome sequences. *PLoS Comput. Biol.* **1:** e22.
Rafalski, A. and Morgante, M. 2004. Corn and humans: Recombination and linkage disequilibrium in two genomes of similar size. *Trends Genet.* **20:** 103–111.
Sauer, S., Lechner, D., Berlin, K., Lehrach, H., Escary, J.L., Fox, N., Gut, I.G. 2000. A novel procedure for efficient genotyping of single nucleotide polymorphisms. *Nucleic Acids Res.* **28:** E13.
Sanchez-Moran, E., Armstrong, S.J., Santos, J.L., Franklin, F.C., and Jones, G.H. 2002. Variation in chiasma frequency among eight accessions of *Arabidopsis thaliana*. *Genetics* **162:** 1415–1422.
Schmidt, R., West, J., Love, K., Lenehan, Z., Lister, C., Thompson, H., Bouchez, D., and Dean, C. 1995. Physical map and organization of *Arabidopsis thaliana* chromosome 4. *Science* **270:** 480–483.
Schnable, P.S., Hsia, A.P., and Nikolau, B.J. 1998. Genetic recombination in plants. *Curr. Opin. Plant Biol.* **1:** 123–129.
Stahl, F.W., Foss, H.M., Young, L.S., Borts, R.H., Abdullah, M.F., and Copenhaver, G.P. 2004. Does crossover interference count in *Saccharomyces cerevisiae*? *Genetics* **168:** 35–48.
Takano-Shimizu, T. 2001. Local changes in GC/AT substitution biases and in crossover frequencies on *Drosophila* chromosomes. *Mol. Biol. Evol.* **18:** 606–619.
Wright, S.I., Agrawal, N., and Bureau, T.E. 2003. Effects of recombination rate and gene density on transposable element distributions in *Arabidopsis thaliana*. *Genome Res.* **13:** 1897–1903.
Yao, H., Zhou, Q., Li, J., Smith, H., Yandeau, M., Nikolau, B.J., and Schnable, P.S. 2002. Molecular characterization of meiotic recombination across the 140-kb multigenic a1-sh2 interval of maize. *Proc. Natl. Acad. Sci.* **16:** 16.
Yu, A., Zhao, C.F., Fan, Y., Jang, W.H., Mungall, A.J., Deloukas, P., Olsen, A., Doggett, N.A., Ghebranious, N., Broman, K.W., et al. 2001. Comparison of human genetic and sequence-based physical maps. *Nature* **409:** 951–953.
Zickler, D. and Kleckner, N. 1999. Meiotic chromosomes: Integrating structure and function. *Annu. Rev. Genet.* **33:** 603–754.

Web site references

http://www.arabidopsis.org/; The Arabidopsis information resource.
http://ftp.tigr.org/pub/data/a_thaliana/ath1/; Arabidopsis sequence database.
http://cran.r-project.org; R software.

Received June 20, 2005; accepted in revised form September 28, 2005.

ANNEXE D

Etude statistique de la coopération entre composantes de réseaux biologiques, *Transactions on Computational Systems Biology*

Depuis une décennie, la disponibilité en quantité croissante de données expérimentales d'interaction entre les composants d'une cellule a permis de reconstruire des réseaux d'interaction *entre ces composantes. Ces réseaux sont représentés formellement par des graphes dont les nœuds sont les composantes, et les arêtes représentent l'existence d'une interaction entre les nœuds qu'elles relient. Divers types d'interaction ont pu être abstraits sous cette forme : interaction physique entre protéines, interactions régulatoires (par exemple, le produit d'un gène active l'expression d'un autre gène), interactions métaboliques (une protéine impliquée dans la métabolisation d'un produit d'une réaction catalysée par une autre protéine)... Ces diverses interactions décrivent des aspects différents, mais pas indépendants, de la vie de la cellule.*

Aussi, comprendre comment ces composantes coopèrent peut permettre de mieux appréhender la manière dont, par exemple, la régulation adapte *le métabolisme d'un organisme en fonction de l'environnement qu'il rencontre. Dans le cas présenté ici,* S. SMIDTAS *avait reconstruit, dans le cas de la levure, un réseau d'interactions protéines/protéines, ainsi qu'un réseau de régulation. Le premier s'entend comme une représentation des interactions entre les protéines permettant au métabolisme d'opérer, alors que le second conditionne l'expression des protéines impliquées dans le premier à un moment donné. Par ailleurs, ces deux réseaux sont liés par une* interface*, composée des protéines présentant des interactions avec d'autres protéines, tout en étant facteur ou cible de la régulation. transcriptionnelle.*

Le projet poursuivi recherchait une caractérisation de cette interface, en particulier vis-à-vis de la topologie du réseau issu de la fusion des deux réseaux d'interaction étudiés. La propriété topologique étudiée était ici la distance (en nombre d'arêtes composant le plus cours chemin d'une protéine à une autre, quelque soit la nature - interactions de protéines ou interaction transcriptionnelle - de cette arête, et en respectant le caractère orienté des arêtes représentant les interactions transcriptionnelles) séparant deux protéines impliquées dans ces interactions. Afin de caractériser cette interface tout en prenant en compte la structure propre de chacun des réseaux qu'elle joint, il a été choisi d'étudier l'impact d'un choix différent de l'interface. Pour ce faire, un grand nombre de réseaux obtenus par un recollement *au hasard des deux réseaux a été généré, et la distribution des histogrammes des distances entre protéines ainsi obtenue a permis de faire apparaître des propriétés saillantes de l'interface réelle entre ces deux réseaux.*

Ce travail a fait l'objet d'une publication dans Transactions on Computational Systems Biology.

Property-driven statistics of biological networks

Pierre-Yves Bourguignon[1], Vincent Danos[2], François Képes[3], Serge Smidtas[1], and Vincent Schächter[1]

[1] Genoscope
[2] CNRS & Université Paris VII
[3] CNRS

Abstract. An analysis of heterogeneous biological networks based on randomizations that preserve the structure of component subgraphs is introduced and applied to the yeast protein-protein interaction and transcriptional regulation network. Shuffling this network, under the constraint that the transcriptional and protein-protein interaction subnetworks are preserved reveals statistically significant properties with potential biological relevance. Within the population of networks which embed the same two original component networks, the real one exhibits simultaneously higher bi-connectivity (the number of pairs of nodes which are connected using both subnetworks), and higher distances. Moreover, using restricted forms of shuffling that preserve the interface between component networks, we show that these two properties are independent: restricted shuffles tend to be more compact, yet do not lose any bi-connectivity.
Finally, we propose an interpretation of the above properties in terms of the signalling capabilities of the underlying network.

1 Introduction

The availability of genome-scale metabolic, protein-protein interaction and regulatory networks [25, 7, 3, 5, 21] —following closely the availability of large graphs derived from the Internet hardware and software network structure, from social or collaborative relationships— has spurred considerable interest in the empirical study of the statistical properties of these 'real-world' networks. As part of a wider effort to reverse-engineer biological networks, recent studies have focused on identifying *salient* graph properties that can be interpreted as 'traces' of underlying biological mechanisms, shedding light either on their dynamics [23, 11, 6, 28] (*i.e.*, how the connectivity structure of the biological process reflects its dynamics), on their evolution [10, 30, 27] (*i.e.*, likely scenarios for the evolution of a network exhibiting the observed property or properties), or both [9, 14, 15]. The statistical graph properties that have been studied in this context include the distribution of vertex degrees [10, 9], the distribution of the clustering coefficient and other notions of density [17–19, 22, 4], the distribution of vertex-vertex distances [22], and more recently the distribution of network motifs occurrences [15].

Identification of a salient property in an empirical graph —for example the fact that the graph exhibits a unexpectedly skewed vertex degree distribution— requires a prior notion of the distribution of that property in a class of graphs relatively to which saliency is determined. The approach chosen by most authors so far has been to use a *random graph model*, typically given by a probabilistic graph generation algorithm that constructs graphs by local addition of vertices and edges [20, 1, 24]. For the simplest random graph models, such as the classical Erdös-Rényi model (where each pair of vertices is connected with constant probability p, [2]), analytical derivations of the simplest of the above graph properties are known [20, 1].

In the general case, however, analytical derivation is beyond the reach of current mathematical knowledge and one has to retort to numerical simulation. The random graph model is used

to generate a sample of the corresponding class of graphs and the distribution of the graph property of interest is evaluated on that sample, providing a standard against which the bias of the studied graph can be measured [23, 14, 29]. Perhaps because of the local nature of the random graph generation process, it is mostly simple *local* network properties that have been successfully reproduced in that fashion. Another, somewhat more empirical, category of approaches reverses the process: variants are generated from the network of interest using a random rewiring procedure. The procedure selects and moves edges randomly, preserving the global number of edges, and optionally their type, as well as local properties such as the degree of each vertex. Rewirings are thus heuristic procedures which perform a sequence of local modifications on the structure of the network.

The specific focus of the present paper is on measuring the degree of cooperation between the two subgraphs of the yeast graph of interactions induced by the natural partition of edges as corresponding either to transcriptional interaction (directed) or to protein protein interaction (undirected). To evaluate a potential deviation with respect to such a measure, one needs as a first ingredient a suitable notion of random variation of the original graph. The goal is here, as in many other cases, to contrast values of a given observable on the real graph, against the distribution of those same values in the population of variants. We define *shuffles* of the original graph as those graphs that are composed exactly of the original two subgraphs of interest, the variable part being the way these are 'glued' together.

From the probabilistic point of view, this notion of randomisation coincides with a traditional Erdös-Renyi statistics, except that it is conditioned by the preservation of the original subgraphs. Designing a generative random graph model that would only yield networks preserving this very precise property seems to be a hard endeavor ; it is not as easy as in the unconditional Erdös-Renyi model to draw edges step by step yet ensure that component subgraphs will be obtained in the end. Shuffling might also be seen as rewiring, except the invariant is large-scale and extremely precise: it is not edges that are moved around but entire subgraphs. Moving edges independently would break the structure of the subnetworks, and designing a sequential rewiring procedure that eventually recovers that structure is not an obvious task. Moreover, it would be in general difficult to ensure the uniformity of the sample ; see [16] for a thorough analysis of rewiring procedures. This choice of an invariant seems rather natural in that one is interested in qualifying the interplay between the original subgraphs in the original graph. Now, it is not enough to have a sensible notion of randomisation, it is also crucial to have a computational handle on it. Indeed, whatever the observable one wants to use to mark cooperation is, there is little hope of obtaining an analytic expression for its distribution, hence one needs sampling. Fortunately, it turns out it is easy to generate shuffles uniformly, since these can be described by pairs of permutations over nodes, so that one can always sample this distribution for want of an exact expression. As explained below in more details, the analysis will use two different notions of subgraph-preserving sampling: *general* shuffles, and *equatorial* ones that also preserve the interface between our two subgraphs. Equatorial shuffles are feasible as well, and in both cases the algorithms for sampling and evaluating our measures turn out to be fast enough so that one can sweep over a not so small subset of the total population of samples.

Regarding the second necessary ingredient, namely which observable to use to measure in a meaningful way the otherwise quite vague notion of cooperation, there are again various possibilities. We use two such observables in the present study: the *connectivity*, defined as the percentage of disconnected pairs of nodes, and a refined quantitative version of connectivity, namely the full distance distribution between pairs of nodes. The latter is costlier, requiring about three hours of computation for each sample on a standard personal computer.

Once we have both our notion of randomisations and our observables in place, together with a feasible way of sampling the distribution of the latter, we can start. Specifically we run four

experiments, using general or equatorial shuffling, and crude or refined connectivity measures. The sampling process allows us to compare the values of these measures for the original graph with the mean value for the sample, and, based on the assumption that those values follow a normal distribution over the sample, one can also provide a p-value that gives a rough estimate of the statistical deviation of the observable in the given graph.

The general shuffle based experiments show with significant statistical confidence that shuffling reduces connectivity (1), and at the same time contracts distances (2). More precisely, both bi-connectivity (the amount of pairs of nodes which are connected using both subgraphs) and distances are higher than average in the real network. A first interpretation might be that the real graph is trading off compactness for better bi-connectivity. In order to obtain a clearer picture and test this interpretation, we perform two other experiments using equatorial shuffles. Surprisingly, under equatorial shuffles connectivity hardly changes, while the global shift to shorter distances is still manifest. It seems therefore there is actually no trade-off, and both properties (1) and (2) have to be thought of as being independently captured by the real graph. With appropriate caution, we may try to provide a biological interpretation of this phenomenon. Since all notions of connectivity and distances are understood as directed, we propose to relate this to signalling, and interpret bi-connectivity as a measure of the capability to convey a signal between subgraphs. With this interpretation, the above properties may be read as:(1) signal flows better than average and (2) signal is more specific than average. The second point requires explanation. At constant bi-connectivity, longer average distances imply that upon receipt of a signal, the receiver has a better chance of guessing the emitter. In other words, contraction of distances (which can be easily achieved by using hubs) will anonymise signals, clearly not a desirable feature in a regulatory network. Of course this is only part of the story, since some hubs will also have an active role in signal integration and decision making. The latter is probably an incentive for compactness. If our reading of the results is on track, we then may think of the above experiments as showing that the tropism to compactness due to the need for signal integration, is weaker than the one needed for signal specificity.

Beyond the particular example we chose to develop here because of the wealth of knowledge available on the yeast regulatory and protein interaction networks, one can think of many other applications of the shuffling methodology for heterogeneous networks. The analyses performed here rely on edges corresponding to different types of experimental measurements, but edges could also represent different types of predicted functional links. Indeed, there are many situations where a biological network of interactions can be naturally seen as heterogeneous. Besides, the notions of shuffle we propose can also accomodate the case where one would use a partition of nodes, perhaps given by clustering, or localisation, or indeed any relevant biological information, and they may therefore prove useful in other scenarios.

The paper is organised as follows: first, we set up the definitions of edge-based general and equatorial shuffles based, and also consider briefly node-based shuffles though these are not used in the sequel; then we describe the interaction network of interest and the way it was obtained; finally we define our observables and experiments, and interpret them. In the conclusion, we discuss generalization and potential applications of the method. The paper ends with an appendix on the algorithmical aspects of the experiments, and a brief recall of the elementary notions of statistics we use to assert their significance.

2 Shuffles

Let $G = (V, E)$ be a directed graph, where V is a finite set of nodes, and E is a finite set of directed edges over V. We write M for the incidence matrix associated to G. Since G is directed,

M may not be symmetric. In the absence of parallel edges M has coefficients in $\{0,1\}$, where parallel edges are allowed.

Given such a matrix M and a permutation σ over V, one writes $M\sigma$ for the matrix defined as for all u, v in V:

$$M\sigma(u,v) := M(\sigma^{-1}u, \sigma^{-1}v)$$

Note that $M\sigma$ defines the same abstract graph as M does, since all σ does is changing the nodes names.

2.1 Shuffles Induced by Properties on Edges

We consider first shuffles induced by properties on edges. Suppose given a partition of $E = \sum E_i$; this is equivalent to giving a map $\kappa : E \to \{1, \ldots, p\}$ which one can think of as colouring edges.

Define M_i as the incidence matrix over V containing the edges in E_i (of colour i).

Define also V_I, where $I \subseteq \{1, \ldots, p\}$, as the subset of nodes v having for each $i \in I$ at least one edge incident to v with colour i, and no incident edge coloured j for $j \notin I$. We abuse notation and still write $\kappa(u) = I$ when $u \in V_I$. This represents the set of colours seen by the nodes.

Clearly $V = \sum V_I$, $V_\varnothing$ is the set of isolated nodes of G, and the set of nodes of G_i is the union of the graphs generated by V_I, for $i \in I$.

Given $\sigma_1, \ldots, \sigma_p$ permutations over V, define the *global shuffle* of M as:

$$M(\sigma_1, \ldots, \sigma_p) := \textstyle\sum_i M_i\sigma_i$$

The preceding definition of $M\sigma$ is the particular case where $p = 1$ (one has only one colour common to all edges). Each G_i (the abstract graph associated to M_i) is preserved up to isomorphism under this transformation. However the way the G_is are glued together is not, since one uses a different local shuffle on each.

For moral comfort, we can check that any means of glueing together the G_is is obtainable using a general shuffle in the following sense: given G' and $\sum q_i : \sum G_i \to G'$ where the disjoint sum $\sum_i q_i$ is an isomorphism on edges, one has that G' is a general shuffle of G. To see this, define $\sigma_i(u) := q_i p_i^{-1}(u)$ if $u \in \kappa^{-1}(i)$, $\sigma_i(u) = u$ else (we have written p_i for the inclusion of G_i in G), one then has $G' = \sum G_i\sigma_i = G(\sigma_1, \ldots, \sigma_p)$.

Note also that $(M(\sigma_1, \ldots, \sigma_p))\tau := \sum_i M_i(\tau\sigma_i)$, and so in particular, without loss of generality one can take any the σ_i's to be the identity (just take $\tau = \sigma_i^{-1}$). This is useful when doing actual computations, and avoids some redundancy in generating samples.

An additional definition will help us refine the typology of shuffles. One says a shuffle $M\sigma$ is *equatorial* if in addition for all I, and all i, V_I is closed under σ_i. Equivalently, one can ask that $\kappa \circ \sigma_i = \kappa$. An *equatorial shuffle* preserves the set of colours associated with each node and in particular preserves for a given pair of nodes (u, v) the fact that (u, v) is heterochromatic, *i.e.*, $\kappa(u) \cap \kappa(v) = \varnothing$. This in turn implies that the distance between u and v must be realised by a path which uses edges of different colours. In the application such paths are mixing different types of interaction, and are therefore of particular interest; without preserving this attribute, an observable based on path with different colours would not make sense. In the particular case of two colours, nodes at the 'equator', having both colours, will be globally preserved, hence the name.

2.2 Shuffles Induced by Properties on Vertices

One can also consider briefly shuffles induced by properties on nodes. Suppose then given a partition of nodes $V = \sum_i V_i$, again that can be thought of as a colouring of nodes $\kappa : V \to \{1, \ldots, p\}$, and extended naturally to the assignment of one or two colours to each edge.

A node shuffle is defined as a shuffle associated to σ which can be decomposed as $\sum_i \sigma_i$, σ_i being a permutation over each cluster V_i. Clearly each graph G_i generated by V_i is invariant under the transformation: only the inter-cluster connectivity is modified.

The equivalent of the equatorial constraint would be to require in addition $\sigma(u) \in \partial V_i$ if $u \in \partial V_i$, where ∂V_i is defined as those nodes of V_i with an edge to some V_j, $i \neq j$. Other variants are possible and the choice of the specific variant will likely depend on the particular case study. We now turn to the description of the network the shuffle experiments will be applied to.

3 A Combined Network of Regulatory and Protein-Protein Interactions in Yeast

With our definitions in place, we can now illustrate the approach on a heterogeneous network obtained by glueing two component networks.

It is known that regulatory influences, including those inferred from expression data analysis or genetic experiments, are implemented by the cell through a combination of direct regulatory interactions and protein-protein interactions, which propagate signals and modulate the activity level of transcription factors. The detailed principles underlying that implementation are not well understood, but one guiding property is the fact that protein interaction and transcriptional regulation events take place in the regulatory network at different time-scales.

In order to clarify the interplay between these two types of interactions, we have combined protein-protein (PPI) and protein-DNA (TRI, for 'transcriptional regulation interaction') interaction data coming from various sources into a heterogeneous network by glueing together these two networks on the underlying set of yeast proteins.

The data from which the composite network was built includes: 1440 protein complexes identified from the literature, through HMS-PCI or TAP [3, 5], 8531 physical interactions generated using high-throughput Y2H assays [26], and 7455 direct regulatory interactions compiled from literature and from ChIP-Chip experiments [4, 12], connecting a total of 6541 yeast proteins. A subnetwork of high-reliability interactions was selected, using a threshold on the confidence levels associated to each inferred interaction. For the ChIP-Chip data produced by Lee et al. [12], interactions with a p-value inferior to 3.10^{-2} were conserved ; for the Y2H data produced by Ito et al. [26], a threshold of 4.5 on the Interaction Sequence Tag was used (see [8]). The PPI network was built by connecting two proteins, in both directions, whenever there was a protein-protein or a complex interaction between the two corresponding proteins. In the case of the TRI network, an edge connects a regulator protein with its regulatee. To simplify the discussion, we will refer in the rest of the paper to the TRI graph as TRI, and to the PPI graph as PPI. With some more precision, define G as the real graph, TRI as the subgraph induced by the set of TRI nodes, *i.e.,* nodes such that $TRI \in \kappa(u)$, and PPI as the subgraph induced by the set of PPI nodes.

Their respective sizes are:

$$TRI = 3387,\ PPI = 2517,\ TRI \cup PPI = 4489,\ TRI \cap PPI = 1415$$

The set of nodes $TRI \cap PPI$ of both colours is also referred to in the sequel as the *equator* or the *interface*. Since the object of the following is to discuss the interplay between the TRI and PPI subgraphs, the interface naturally plays an important role. A qualitative measure of the connectivity between TRI and PPI which will be useful later in the discussion, is the number of bi-connected pairs in G (these are the pairs which are connected in G, but not connected in either TRI or PPI), which is roughly $p_{bi} = 23\%$. To complete this statistical portrait of the data, we provide in figure 1 the histograms of degree distributions in the PPI and TRI networks, with in and out degrees pictured separately for the latter. Figure 1 also shows the hub size distribution

for the TRI network (the PPI network has no non-trivial hubs). Note that hubs are defined as sets of nodes connected to a single node. The TRI network (here considered as unoriented) has 124 such hubs ; the histogram of the distribution of their sizes is given in figure 1.

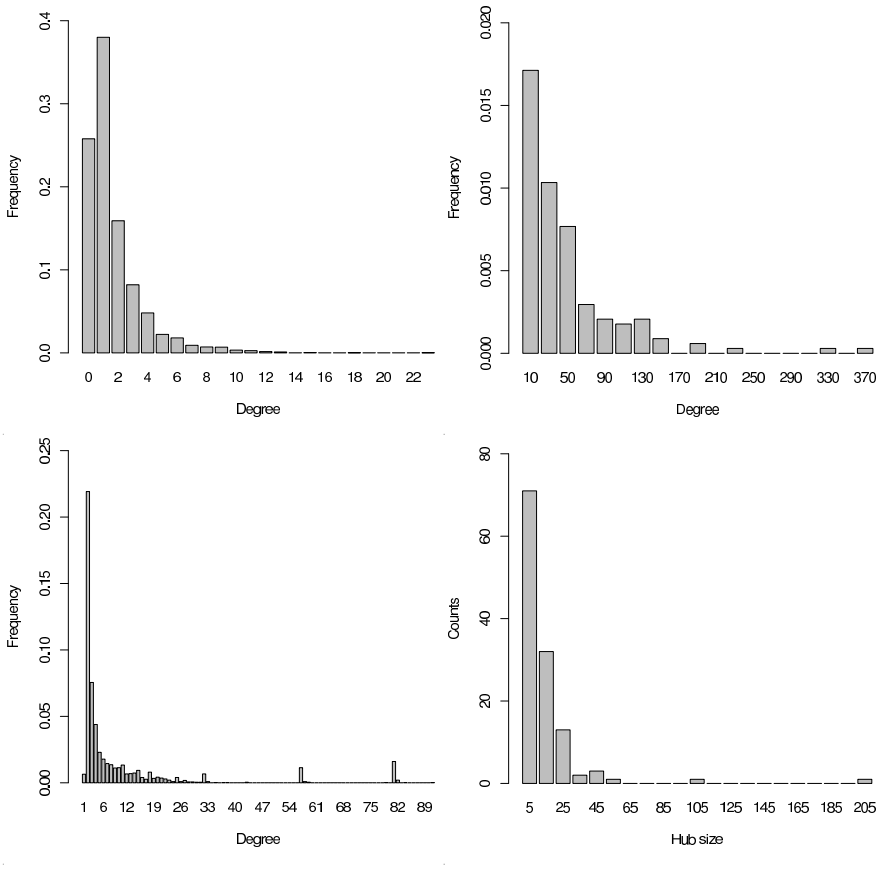


Fig. 1. First row: Histograms of the in and out degree distributions of the TRI network. Second row: Histogram of the degree distribution of the PPI network and of the distribution of the hub size in the TRI network.

4 Results and Interpretations

Hereafter, notions of connectivity, distance, etc. should be understood as *directed* unless explicitly stated otherwise. We now turn to the various shuffle experiments and consecutive observations.

4.1 General Shuffle vs Connectivity

We take here as a rough measure of the connectivity of a graph the percentage of unconnected pairs. Comparing first the real graph with the randomised versions under the general shuffle, one

finds that in the average 4% of the population pairs are disconnected under shuffle. So general shuffle disconnects, or in other words G maximises bi-connectivity.

Clearly mono-connected pairs (pairs connected in either PPI or TRI) cannot be disconnected under general shuffle; a pair is 'breakable' only if bi-connected in G; therefore a more accurate measure of the connectivity loss under general shuffle is that about 17.5% of the breakable pairs are actually broken (this obtained by dividing by p_{bi}), a rather strong deviation with a p-value below 10^{-11}.

Inasmuch as a directed path can be thought of as a signal-carrying pathway, one can interpret the above as saying that the real graph connects PPI and TRI so as to maximise the bandwidth between the subgraphs.

4.2 Equatorial Shuffle vs Connectivity

Keeping with the same observable, we now restrict to equatorial shuffles. One sees in this case that no disconnection happens, and actually about 1% more pairs are connected *after* shuffling. The default of connected pairs of the real graph has a far less significant p-value of 3%. However the point is that equatorial shuffles leaves bi-connectivity rather the same.

This complements the first observation and essentially says that the connectivity maximisation seen above is a property of the set of equatorial nodes ({TRI,PPI} nodes) itself, and not of the precise way TRI and PPI edges meet at the equator.

Both observations can be understood as saying that the restriction of G to the equator is a much denser subgraph than its complement (as evidenced by the connectivity loss under general shuffle), and dense enough so that equatorial shuffling does not impact connectivity.

Note that so far the observable is somewhat qualitative, being only about whether a pair is connected or not. Using a refined and quantitative version of connectivity, namely the distribution of distances (meaning for each n the proportion of pairs at distance n), will reveal more.

4.3 Impact of Equatorial Shuffles on Distance Distribution

Using this refined observable, one sees that the whole histogram shifts to the left, so equatorial shuffle contracts the graph (Fig. 2). This is confirmed by the equality between the number of lost pairs at distance 7 to 9 and the number of new ones at distance 3 to 5. In accordance with the preceding experiment, one also does not see any disconnection under equatorial shuffle.

This is to be compared with the general shuffle version (Fig. 3) where both effects are mixed, and the cumulated excess of short pairs does not account for the loss of long pairs (indeed we know 4% are broken, *i.e.*, disappear at infinity and are not shown on the histogram).

To summarize the distance distribution results in a single number, one can compute the deviation of the real graph mean distance under both shuffles. As expected the mean distance is higher in the real graph with respective p-values of 0.2% and 2% in the general and equatorial shuffles (see Appendix for details). We conclude that while the real graph does maximise bi-connectivity, it does not try to minimise the associated distances.

To provide an intuition on the potential interpretation of the above result, let us again consider paths as rough approximations of signalling pathways. Now compare a completely linear chain-shaped graph and star-shaped one, with the same number of nodes and edges. In the star case, any two nodes are close, at constant distance 2, while in the chain distances are longer. As said, compactness comes with a price, namely that in a star graph all signals go through the hub and are anonymised, *i.e.*, there may be a signal, but there is no information whatsoever in the signal about where the signal originated from. Quite the opposite happens in a linear graph. Of course this is an idealized version of the real situation; nevertheless it is tempting to interpret this last

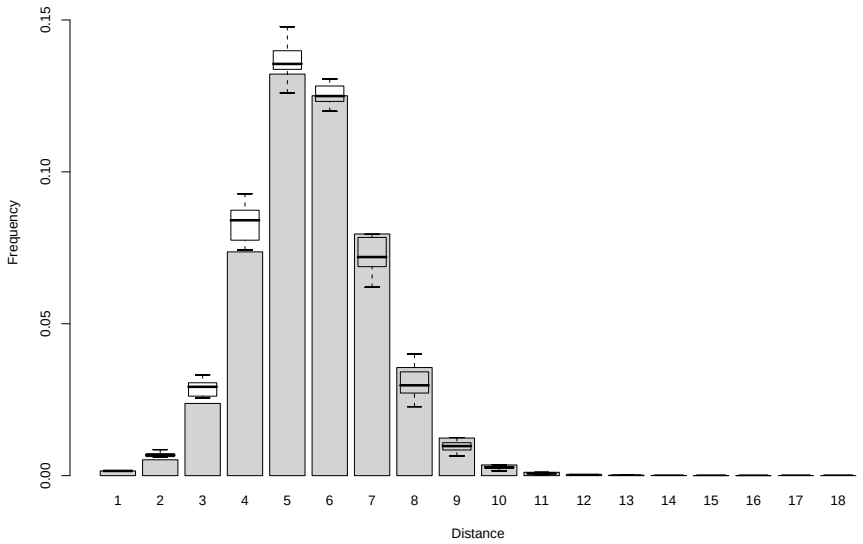


Fig. 2. Equatorial shuffle distance histogram: grey boxes stand for the real graph; one sees that shuffles have more pairs at shorter distance, and consequently (because the number of connected pairs is about the same) less such at higher distances.

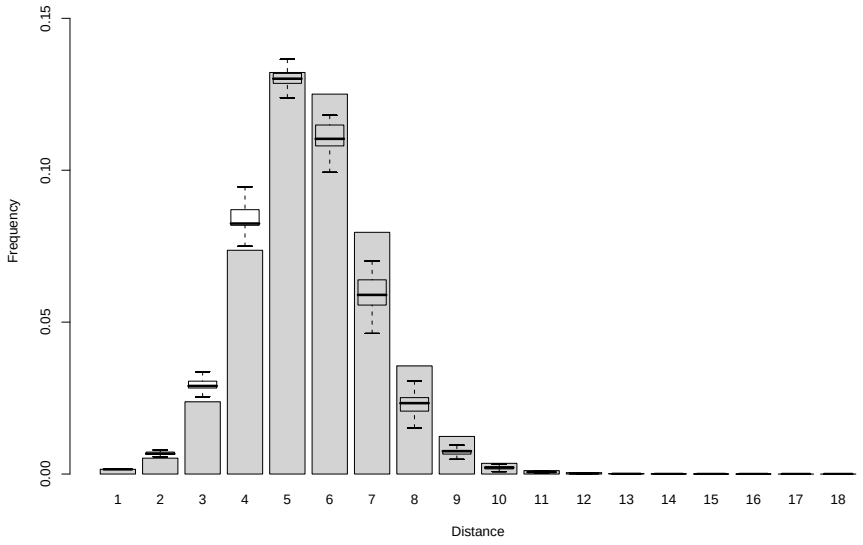


Fig. 3. Global shuffle distance histogram

observation as an indication that the real graph is trading off fast connectivity against specificity of signals. The heterogeneous network is likely to result from a trade-off between causality and signal integration.

As suggested in the introduction, finer observables would have to be developed to further refine this interpretation. Furthermore, there are intrinsic limits on the nature of properties that can be identified using pure topology; deeper, reliable insights about signal transmission in the joint network will ultimately require a dynamical view of signaling with corresponding experimental data.

5 Conclusion

In order to assess the cooperation between the network of protein-protein interactions and the regulatory network in yeast, we have defined two notions of shuffle, *i.e.* tractable randomisations of the original network that preserve global invariants. While general shuffles preserve the entire structure of the component subnetworks, equatorial shuffles also preserve the interface between the networks. We assessed cooperation between the subnetworks using two observables: the percentage of connected node pairs, and the distribution of distances between nodes. For each shuffle-observable pair, the observable in the real network was assessed against the distribution of observables in the set of network variants generated by the respective shuffle.

To summarise the results of this case study, we can say that the statistical analysis of G shuffles under the constraint of preserving its component subnetworks suggests the existence of two *independent* properties of G regarding the cooperation of its components:

- bi-connectivity, i.e. the proportion of node pairs connected only by paths using both types of edges, is higher in the actual network than in the shuffles;
- distances between pairs of nodes are higher in the actual network;

The first property can be given an interpretation in terms of bandwidth: signals flow better between the two networks than would be expected if they were connected randomly. The second property can be interpreted as favoring signal specificity: for cellular interaction networks (in contrast with telecommunication networks, for instance, where each packet carries significant intrinsic information) the information borne by a signal is very much related to the path it has followed. Longer paths thus provide more opportunity for specific signals. Note that the fact that we worked with directed notions (and not with undirected ones as we did in a first version of this paper) makes the interpretation of paths as potential signaling pathways somewhat more convincing.

We have been careful in the discussion of the results of our statistical experiments in terms of signalling capacities, and this needs to be thrashed out in subsequent work. To do so one would first need refined and yet feasible observables pertaining to the dynamics of the network of interest. A recent paper equips the subgraph induced by the major molecular players in the budding yeast cell cycle (cyclins, their inhibitors, and major complexes) with a discrete Boolean dynamics [13], and obtains a dynamics with a stable state corresponding to the G_1 phase, which is attracting a significantly higher number of states than a random graph (with the same number of nodes and edges) would. It seems therefore possible to explicitly construct signal-related observables. However there are several problems: first, this analysis relies on sorting positive and negative regulation edges, and that is an information which one doesn't have for the full graph; second it also critically relies on the rather small size of the subgraph; finally the model only handles a limited number of signals (corresponding to the various cell cycle phases). Nevertheless, a comparable study, using shuffles as a means of randomising, and confined to a

well-chosen subgraph could help in qualifying our speculative interpretation of the contraction phenomenon we have observed.

On the methodological front, both the general notion of shuffle and the restricted notion of equatorial shuffle proved useful: they reveal different properties and complement one another. The same holds for the pair of observables: both the qualitative connectivity observable and its refined distance-based version are useful, and yield different and complementary insights on the cooperation between the two component networks.

We believe that the shuffling methodology developed for this case study has general applicability to the study of heterogeneous biological networks, i.e. networks that can be seen as the "glueing" of two or more component networks. Shuffles preserve global invariant properties (the structure of component networks), and define rigorously and unambiguously the class of networks which obey these properties. They are also easily computable and can be generated uniformly, by drawing from a set of acceptable permutations. Note that the latter property is in contrast with randomizations based on sequential rewiring strategies, where each rewiring step perturbs the structure while preserving one or more local invariants. While these approaches may prove to be asymptotically equivalent in some cases, they typically do not provide a direct definition nor the means to uniformly sample the set of randomizations which preserve the invariant, since the order of the rewiring steps matters.

Given an interaction network between biochemical species, any biological property on edges (type of interaction, degree of confidence, localization of interaction...) or on nodes (type of entity, functional annotation, inclusion within clusters generated using a given data type and methodology, etc...) with discrete values can be used to define a heterogeneous version of that network. Then, either the type of edge shuffles used above, or shuffles preserving other categories of top-down invariants, such as the projection of a network onto a given network of abstract clusters, could be explored. Likewise, a variety of observable properties may be used to investigate cooperation between component subnetworks. Perhaps the foremost promise of the shuffling approach resides in the interplay between different shuffle-observable pairs, which allows an exploratory assessment of cooperation adapted to the heterogeneous network at hand.

References

1. William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *32nd Annual ACM Symposium on Theory of Computing*, pages 171–180, 2000.
2. P. Erdös and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:1761, 1960.
3. AC Gavin, M Bosche, R Krause, P Grandi, M Marzioch, A Bauer, J Schultz, JM Rick, AM Michon, CM Cruciat, M Remor, C Hofert, M Schelder, M Brajenovic, H Ruffner, A Merino, K Klein, M Hudak, D Dickson, T Rudi, V Gnau, A Bauch, S Bastuck, Huhse, C Leutwein, MA Heurtier, RR Copley, A Edelmann, E Querfurth, V Rybin, G Drewes, M Raida, T Bouwmeester, P Bork, B Seraphin, B Kuster, G Neubauer, and G. Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868)(Jan 10):141–7., 2002.
4. N. Guelzim, S. Bottani, P. Bourgine, and F. Kepes. Topological and causal structure of the yeast transcriptional regulatory network. *Nat Genet*, 31(1):60–3, 2002.
5. Y Ho, A Gruhler, A Heilbut, GD Bader, L Moore, SL Adams, A Millar, P Taylor, K Bennett, K Boutilier, L Yang, C Wolting, I Donaldson, S Schandorff, J Shewnarane, M Vo, J Taggart, M Goudreault, B Muskat, C Alfarano, D Dewar, Z Lin, K Michalickova, AR Willems, H Sassi, PA Nielsen, KJ Rasmussen, JR Andersen, LE Johansen, LH Hansen, H Jespersen, A Podtelejnikov, E Nielsen, J Crawford, V Poulsen, BD Sorensen, J Matthiesen, RC Hendrickson, F Gleeson, T Pawson, MF Moran, D Durocher, M Mann, CW Hogue, D Figeys, and M Tyers. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868)(Jan 10):180–3, 2002.

6. J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nat Genet*, 31(4):370–7, 2002.
7. T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci U S A*, 98(8):4569–74, 2001.
8. Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Masahira Yoshida, Mikio a nd Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein inter actome. *PNAS*, 98(8):4569–4574, 2001.
9. H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–2, 2001.
10. H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai, and A.-L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407:651–654, 2000.
11. N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 2004.
12. T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002.
13. Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *PNAS*, 101(14):11250–11255, April 2004.
14. R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–42, 2004.
15. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–7, 2002.
16. R. Milo, S. S. Shen-Orr, S. Itzkowitz, N. Kashtan, D. Chklovskii, and U. Alon. On the uniform generation of random graphs with prescribed degree sequence. *ArXiv*, 2003.
17. M. E. Newman. Assortative mixing in networks. *Phys Rev Lett*, 89(20):208701, 2002.
18. M. E. Newman. Properties of highly clustered networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 68(2 Pt 2):026121, 2003.
19. M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 69(2 Pt 2):026113, 2004.
20. M. E. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys Rev E Stat Nonlin Soft Matter Phys*, 64(2 Pt 2):026118, 2001.
21. N. D. Price, J. A. Papin, C. H. Schilling, and B. O. Palsson. Genome-scale microbial in silico models: the constraints-based approach. *Trends Biotechnol*, 21(4):162–9, 2003.
22. E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–5, 2002.
23. S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nat Genet*, 31(1):64–8, 2002.
24. S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–76, 2001.
25. P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403(6770):623–7, 2000.
26. C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, 2002.
27. A. Wagner. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol Biol Evol*, 18(7):1283–92, 2001.
28. D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–2, 1998.
29. E. Yeger-Lotem, S. Sattath, N. Kashtan, S. Itzkovitz, R. Milo, R. Y. Pinter, U. Alon, and H. Margalit. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proc Natl Acad Sci U S A*, 101(16):5934–9, 2004.
30. S. H. Yook, H. Jeong, A. L. Barabasi, and Y. Tu. Weighted evolving networks. *Phys Rev Lett*, 86(25):5835–8, 2001.

A Computation of the Shortest Paths Length Distribution

This section is devoted to a brief description of the algorithms and methods used to derive the various statistics used in the study of the yeast regulation and protein interaction networks.

Clearly the (i, j) coefficient of M^n is the number of oriented paths of length n connecting i to j in the graph underlying M. Since we are only interested in knowing whether two nodes are connected by an oriented path of a given length we may use a simplified matrix product defined as:

$$M^n(i,j) = \begin{cases} 1 & \text{if } \exists k \in V : M^{n-1}(i,k) = M(k,j) = 1 \\ 0 & \text{otherwise} \end{cases}$$

which is just forgetting the numbers of connecting paths, only to remember whether there is at least one.

Furthermore, the addition of the identity matrix I to the adjacency matrix before the computation of the products gives an immediate access to the value of the cumulative distribution function of the oriented, shortest path length distances in the network. Indeed, writing $\widehat{M} = M + I$:

$$\widehat{M}^n(i,j) = \begin{cases} 1 & \text{if } \exists k \in V, M^{n-1}(i,k) = M(k,j) = 1 \\ & \text{or } \widehat{M}^{n-1}(i,j) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Thus the number of 1s in $\widehat{M}(G)^n$ is the number of ordered pairs connected by at least one path of length $\leq n$, and the entire distribution is obtained when the computation reaches a fixpoint. Computing the distribution on the real PPI-TRI graph takes about 180' on a recent PC ; the distribution for the 100 shuffles were computed down in less than 10 hours on a cluster of 41 computers hosted by Genoscope.

B Statistics

This section details the definition and computation of p-values shown in the statistical results, concerning both the amount of connected pairs and the average distance.

In order to compute p-values for the deviation of the observable on the real graph from its disribution over the set of shuffled ones, we need to approximate this distribution by a Gaussian one, with mean and standard deviation fixed to the empirical values computed on the sample. This is necessary, since the rather low amount of shuffled networks (100) prevents a direct estimation of the p-value as the proportion of shuffled networks with a larger observable.

Concerning the amount of disconnected pairs, which is the first observable considered in the results, the empirical mean over the set of general shuffles is $m_g = 0.574$, and the standard deviation $s_g = 0.005$. In the case of the equatorial shuffle, the mean falls to $m_e = 0.534$, with a standard deviation of 0.002.

Assuming this average proportion is a Gaussian random variable A with those parameters, the p-value of the deviation of the average proportion of disconnected pairs in the real network from its distribution over the sample of general shuffled networks is defined as:

$$p_g = \mathbb{P}(A < m_G), \quad \textit{with } A \sim \mathcal{N}(m_g, s_g)$$

where $m_G = 0.538$ is the observed proportion of disconnected pairs in the real network. In this case, this yields $p_g = 9 \times 10^{-12}$.

Since the proportion of disconnected pairs in the real graph is higher than the average amount of disconnected pairs in the equatorially shuffled ones, one computes the p-value p_e using the upper tail of the distribution instead of the lower one:

$$p_e = \mathbb{P}(A > m_G), \quad with\ A \sim \mathcal{N}(m_e, s_e)$$

so that $p_e = 0.03$.

The computation of the p-value for the deviation of the mean distance from its value on shuffled networks follows the same scheme. The mean distance in the real graph is $m_G^d = 5.66$, while its average over the set of shuffled graphs is $m_g^d = 5.38$ for the general shuffle, and $m_e^d = 5.5$ for the equatorial one. Standard deviation is $s_g^d = 0.09$ with the general shuffle, and $s_e^d = 0.08$ with the equatorial one. The p-values for these deviations are $p_g^d = 0.002$ and $p_e^d = 0.02$, respectively.