Prédiction de la localisation cellulaire des protéines à l'aide de leurs séquences biologiques.

Hugues Richard

UNIVERSITÉ D'EVRY VAL D'ESSONNE
2005

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ D'EVRY VAL D'ESSONNE

Spécialité : Mathématiques appliquées à la biologie

présentée et soutenue publiquement

par

M. Hugues RICHARD

le 15 décembre 2005

Titre :

Prédiction de la localisation cellulaire des protéines à l'aide de leurs séquences biologiques.

Directeurs de Thèse : MM. François Képès & Bernard Prum

JURY

M. Alain Guénoche	Rapporteur
M. Alain Denise	Rapporteur
Mme Marie-Hélène Mucchielli-Giorgi	Examinateur
M. Burkhard Rost	Examinateur
M. Michel Termier	Examinateur

Remerciements

Je tiens principalement à remercier Alain Guénoche et Alain Denise, pour avoir accepté de rapporter ma thèse, et pour toutes les remarques constructives, qu'il ont pu faire sur le manuscrit. Merci aussi à Michel Termier et Burkhard Rost pour leur participation à mon jury en tant qu'examinateurs.

Trois personnes m'ont encadré durant ces années de thèse ; Bernard, dont la disponibilité et la pédagogie légendaires m'ont sauvé plus d'une fois la mise, François dont les explications précises ont toujours éclairci les doutes que j'ai pu avoir concernant certains points biologiques. Marie-Hélène aussi, après avoir initié ce sujet, et malgré de nouveaux projets de recherche à suivre, a toujours su dégager le temps nécessaire pour me guider sur ce sujet difficile. Je vous exprime ici ma gratitude pour les multiples conseils avisés que vous avez su me prodiguer.

J'ai souvent tendance à rendre le lendemain un projet prévu pour l'avant-veille. Merci à vous Catherine, Florence, Cédric et Vincent d'avoir su me rappeler avec justesse et fermeté les échéances calendaires auxquelles j'étais soumis.

Il est difficile de rendre compte d'une ambiance de laboratoire, le plaisir que peuvent avoir les gens à travailler ensemble, ou simplement la bonne humeur émanant des conversations de couloirs. Merci à vous tous du laboratoire Statistique et Génomes : Anne-Sophie, Bernard, Catherine, Cecile, Chanthou, David, Emmanuelle, Etienne, Florence, François, Franck, Grégory, Marc, Marie-Pierre, Maurice, Maxime, Mickael, Nicolas, Pierre, Pierre-Yves, Sophie, Simona, Vincent, pour tous ces moments au laboratoire, les discussions scientifiques bien sûr, mais aussi les grands débats, serrés dans les 10 m^2 de la caféteria, les parties de Quake du vendredi soir ou les pélerinages à Jobim chaque année.

Chanthou, en pratiquant au quotidien la méditation transcendentale dans le capharnaüm de ton bureau, tu as apporté pendant un temps une touche d'entropie amusante.

David, pendant un temps la "force tranquille" du laboratoire, je me souviendrais de toutes nos discussions, les retours en voiture et surtout ta relecture bénéfique du chapitre de classification.

Vincent, toi qui a délaissé vin et charbon pour le calcul parallèle et les grands mystères de la vie, j'ai beaucoup apprécié ces heures penchés ensemble sur d'incompréhensibles lignes de code.

Maurice, avec toi le matérialisme s'est hissé au rang de religion. A mes heures, j'ai essayé d'être un vaillant disciple et je te remercie pour m'avoir toujours permis d'avoir un cran d'avance sur la modernité.

Grégory, grand numéricien -mais jamais numérologue- j'ai eu beaucoup de plaisir à discuter avec toi sur ce joli sujet -bientôt clos par tes soins- qu'est le calcul des significativités de mots.

Cette thèse s'est aussi déroulée au sein du laboratoire de Bioinformatique de Gif-sur-Yvette. Merci donc a Hervé Delacroix d'avoir accepté de m'accueillir dans ce laboratoire et particulièrement à tous les membres de cette équipe pour tous ces moments sympathiques en leur companie. Il était particulièrement agréable de pouvoir arriver dans ce laboratoire en étant accueilli avec gentillesse et simplicité.

Merci aux membres groupe de travail "Statistique des séquences biologiques" pour tous leurs exposés enrichissants.

J'ai aussi eu le plaisir de collaborer avec Michaël Bekaert et Jean-Pierre Rousset sur le sujet de la détection de *ab initio* de sites de frameshift. Merci à eux pour toutes les discussions passionnantes qu'ont suscité ce projet.

Jean-Louis et Lorraine, vous avez accepté, malgré vos maigres connaissances biologiques ou mathématiques de relire tout ou partie de cette thèse. Merci pour vos remarques qui ont permis d'améliorer la clarté et l'orthographe du manuscrit.

Merci à L. Torvald, A. Cox, et à toutes les personnes qui contribuent au système d'exploitation libre GNU Linux que j'ai eu le plaisir d'explorer durant cette thèse ; et merci à Larry Levan pour Perl, le plus esthétiquement confus des languages de programmation qu'il m'ait été donné d'utiliser.

Merci à Jorge Ben, Caetano Veloso, Gilberto Gil et João Gilberto pour avoir ensoleillé de leurs voix chaleureuses la grisaille de ces trajets interurbains quotidiens.

Merci à vous mes parents, et à toi mon cher frérot, pour la chance que j'ai de vous avoir comme famille, et tout ce qui va avec.

Enfin, vous tous, mes amis, merci pour tous ces moments de franche insouciance et de douce gaieté qu'il m'a été donné de partager avec vous entre les lignes de cette thèse.

Un dernier merci à Mélanie et Alexis pour avoir rédigé ces remerciements.

Table des matières

Première partie

Introduction

Préambule

Les compartiments cellulaires, de par les frontières membranaires qui les définissent, permettent l'accomplissement de tâches métaboliques diverses au sein de la cellule. Cette spécialisation en domaines intracellulaires induit donc une différentiation dans la fonction des protéines qui les composent. Ainsi, quand les recherches d'homologues dans les bases de données ne permettent pas de déduire la fonction d'une protéine, la connaissance du compartiment cellulaire où celle-ci réside peut fournir des indices sur sa fonction. Le grand nombre de gènes "orphelins" (*i.e.* sans homologues) produits ces dernières années par les projets de séquençage motive la mise au point de méthodes efficaces pour la prédiction *ab-initio* de la localisation cellulaire des protéines.

Ainsi, la grande majorité de ce travail de thèse s'intéresse au problème de la prédiction du compartiment cellulaire d'une protéine à partir de sa séquence primaire. J'ai aussi utilisé le même type de méthodologies en travaillant sur la séquence nucléique chez la levure.

Le manuscrit est structuré en quatre parties, la première partie rappelant quelques éléments de biologie cellulaire en décrivant les différents niveaux de la compartimentation cellulaire.

La seconde partie introduit les outils mathématiques permettant de se positionner dans un cadre d'étude statistique sur les séquences biologiques. Le chapitre 2 présente ainsi les modèles markoviens et leur utilisation pour la modélisation de séquences. Le chapitre suivant (chapitre 3) présente des éléments de théorie de l'apprentissage statistique, et plus particulièrement une méthode de classification supervisée : les "Support Vector Machine". Ensuite, le chapitre 6 détaille les adaptations nécessaires de ces outils théoriques pour l'application au problème de classification multiclasse qu'est la prédiction du compartiment cellulaire d'une protéine. En particulier, je propose une méthode de classification utilisant un arbre de décision binaire construit à partir de chaînes de Markov cachées estimées sur chaque classe.

La troisième partie détaille la problématique de la prédiction de la localisation cellulaire des protéines. Afin de fixer les idées sur le phénomène biologique étudié, le chapitre 4 insiste sur les hypothèses biologiques faites à la création du jeu de données, et présente ceux que j'utiliserai pour les expérimentations. Après une revue bibliographique des différentes méthodes proposées sur ce sujet (chapitres 5), les deux chapitres

suivants présentent les résultats que j'ai obtenus : en regardant l'influence de l'information présente dans la séquence nucléique pour la discrimination entre certains compartiments (chapitre 7.1), et en utilisant des arbres de décision pour la classification (chapitre 7.2).

Enfin, en annexe est présenté un article décrivant un travail mené parallèlement pendant ma thèse, en collaboration avec des biologistes, où nous avons conçu une stratégie par chaînes de Markov cachées permettant la détection des candidats potentiels au décalage de phase dans le génome de *S. Cerevisiae*.

Chapitre 1

Présentation Biologique

1.1 Vue d'ensemble sur la cellule eucaryote

La cellule est l'unité structurale et fonctionnelle constituant tout ou partie d'un être vivant. Historiquement, ce terme a été introduit en référence à la cellule de moine, et décrit donc un espace clos **isolant** un ensemble de macromolécules du reste du monde. Cette frontière entre milieu intra- et extra- cellulaire est réalisée par la membrane plasmique, constituée majoritairement de protéines et d'une bicouche lipidique. Si les protéines présentes dans et sur la membrane permettent l'interaction de la cellule avec l'extérieur, la bicouche lipidique constitue la frontière à proprement parler. Ainsi, comme on peut le voir sur la figure 1.1, les chaînes hydrophobes (2) sont concentrées au milieu de la membrane, et les têtes polaires (1) isolent les molécules d'eau environnantes.

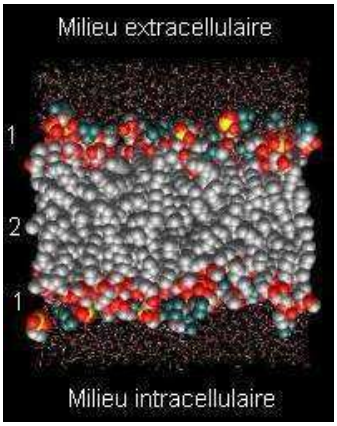


FIG. 1.1 – modèle moléculaire compact d'une bicouche lipidique en milieux aqueux. L'organisation en bicouche fait apparaître une véritable barrière hydrophobe, séparant deux compartiments hydrophiles.

Dans le cas des organismes eucaryotes, des membranes biologiques permettent aussi une compartimentation à l'intérieur de la cellule, prévenant ainsi certaines interactions et en favorisant d'autres.

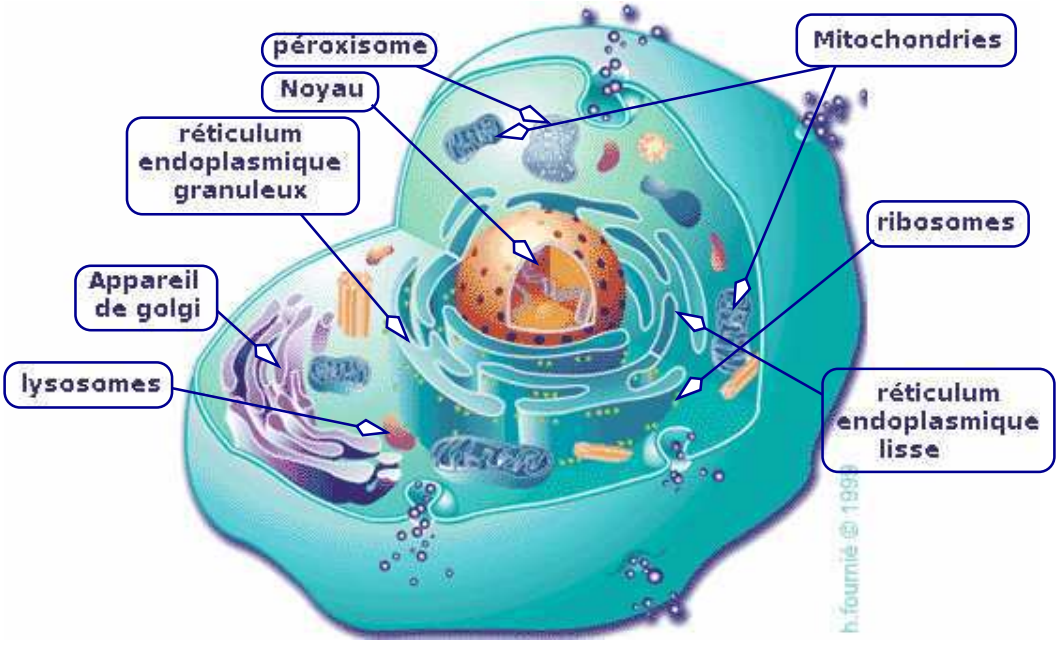


FIG. 1.2 – représentation schématique d'une cellule animale. H. Fournié ©.

La figure 1.2 représente une vue schématique d'une cellule animale. Un certain nombre de membranes définissent ainsi les différentes organites cellulaires, et permettent de dégager quatre grandes catégories fonctionnelles présentes dans la cellule :

- la voie sécrétoire, elle est composée du réticulum endoplasmique (lisse ou granuleux), de l'appareil de Golgi, des vésicules de sécrétion, des lysosomes et des endosomes. Comme on le verra plus loin, ces organites permettent les modifications post traductionnelles sur les protéines, et correspondent à la voie d'exportation principale pour les protéines extracellulaires.
- le noyau, où est stocké l'ADN et où sont assemblés les sous unités des ribosomes. Notons que cet espace n'est pas réellement un compartiment, car il est percé de pores qui permettent une relation directe avec l'espace cytoplasmique.
- les compartiments clos, tels que la mitochondrie et le péroxisome auquel on peut ajouter le chloroplaste dans le cas des cellules végétales.
- le compartiment cytoplasmique, qui contient toutes les structures susnommées, et structuré par un ensemble de polymères protéiques filamenteux, le cytosquelette. Il s'agit aussi du lieu où la majorité des protéines sont synthétisées.

1.2 Compartiments cellulaires et trafic des protéines

En 1999, Günter Blobel obtenait le prix Nobel de médecine pour avoir découvert que “les protéines possèdent des signaux intrinsèques qui gouvernent leur localisation dans la cellule”. En effet, si une cellule eucaryote est compartimentée, des processus permettent d’assurer les échanges protéiques entre les différents compartiments. Ces mécanismes d’adressages permettent ainsi l’import nucléaire des protéines cytosoliques, la translocation des protéines à travers la membrane du réticulum endoplasmique, ou l’import des protéines dans la mitochondrie, le chloroplaste ou le péroxisome.

1.2.1 le noyau

Remarquons que ce compartiment possède comme particularité de n’être défini que lorsque la cellule est en interphase. Il correspond alors à une zone limitée par une région particulière du réticulum endoplasmique alors nommée enveloppe nucléaire.

Rappelons que le noyau est avant tout le lieu de stockage de l’ADN. Celui-ci est structuré en nucléosomes formant la chromatine. Cette information est exploitée par les ARN et ADN polymérases pour la réplication et l’expression génétique. En particulier, la transcription d’une molécule d’ADN en ARN messager a lieu dans le noyau. L’ARN messager nouvellement synthétisé doit ensuite être convoyé hors de la cellule pour être traduit en protéine dans le cytoplasme. De la même manière, les protéines nécessaires à l’expression des gènes (polymérases, facteurs de transcriptions...) doivent être importées dans le noyau après avoir été synthétisées dans le cytoplasme.

Ce transit des macromolécules entre le noyau et le cytoplasme (et inversement) s’opère via les pores nucléaires. Les mécanismes exacts de ces transports ne sont pas complètement connus, cependant, on sait que ce transport ne prend en charge que des protéines. Les ARN sont donc exportés par l’intermédiaire de protéines cargo qui leur sont associées. En outre, le transport met en jeu un système d’adressage basé sur l’existence de séquences spécifiques d’acides aminés. Ainsi, seules les protéines possédant cette étiquette particulière seront transportées. Les séquences les mieux connues sont les séquences NLS (pour Nuclear Localization Signal) caractérisés actuellement par un ou deux motifs composés respectivement d’acides aminés chargés positivement et basiques. Un schéma simplifié illustrant ce fonctionnement est présenté en figure 1.3

Les nucléoles, même s’ils ne forment pas à proprement parler de compartiment, *i.e.* ne sont pas limités par une membrane, méritent d’être cités comme constituants particuliers du noyau. Ces zones ont d’abord été identifiées parce qu’elles sont fortement colorées en microscopie optique. Dans ces zones, une partie des sous unités des ARN ribosomiques (ARNr 45S, précurseur des ARNr 18S, 28S et 5,8S) sont transcrites et maturées à fort régime à partir de plusieurs gènes répétés. Les différentes sous unités sont ensuite assemblées dans le cytoplasme.

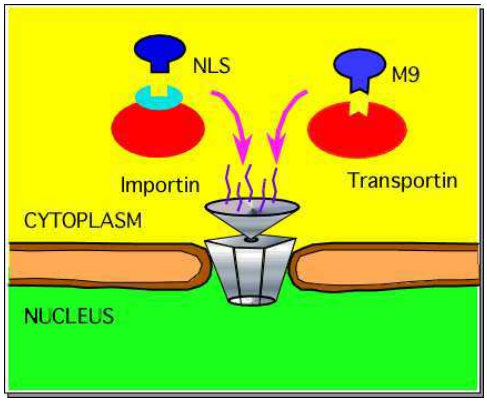


FIG. 1.3 – Schéma simplifié du mécanisme d'import nucléaire. Après synthèse de la protéine nucléaire dans le cytoplasme, une protéine de la famille des importines ou transportines s'accroche sur le signal NLS. Le complexe importin/protéine-NLS (ou transportine/protéines) est ensuite transportée dans le noyau à travers les pores nucléaires.

1.2.2 la voie sécrétoire

Les protéines membranaires et les protéines exportées sont assemblées dans le cytoplasme. Cependant, elles traversent la membrane du réticulum endoplasmique granuleux (REG) pendant leur synthèse. Leur transport entre les organites est ensuite assuré par des événements successifs de fusion et de fission membranaire (figure 1.5) et s'accompagne de modifications post-traductionnelles.

1.2.2.1 le réticulum endoplasmique

Le réticulum endoplasmique peut être décomposé en deux zones en continuité, le réticulum endoplasmique lisse (REL) et le réticulum endoplasmique granuleux (REG, en raison des ribosomes accrochés à la membrane, visibles en microscopie électronique).

Le REL est le siège de l'assemblage des phospholipides membranaires. C'est donc la source de membrane pour la cellule.

L'import d'une protéine dans le REG se fait **conjointement** à sa traduction, et est rendu possible par la présence, à son extrémité N-terminale d'une séquence particulière permettant son adressage. Cette séquence, appelée le **signal peptide**, est constituée de 10 à 25 acides aminés hydrophobes, et est reconnue dès sa sortie du ribosome par un complexe ribonucléoprotéique, la SRP (Signal Recognition Particle). La SRP se fixe sur la séquence signal et provoque un arrêt de la traduction. Cet arrêt permet l'accrochage du complexe ribosome/ARNm/SRP à un récepteur à SRP sur la membrane du REG. Ce récepteur favorise ensuite l'interaction du complexe avec le translocon, un complexe protéique de la membrane du REG, qui forme alors un canal plus ou moins hydrophile permettant le passage de la protéine en synthèse (figure 1.4). Après translocation de la

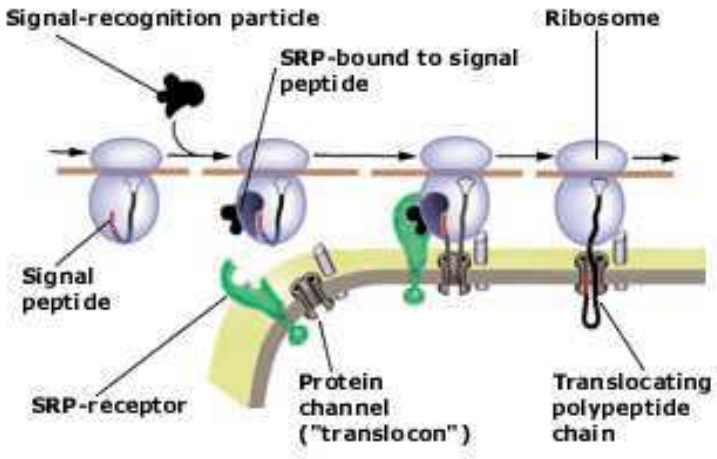


FIG. 1.4 – Vue schématique du mécanisme de translocation d'une protéine à travers la membrane du REG. Après la synthèse complète de la protéine, le signal peptide (rouge) sera clivé.

protéine dans le REG, la séquence signal est clivée en un site spécifique situé après le signal peptide.

Les protéines membranaires sont adressées suivant le même mécanisme, chaque segment transmembranaire étant intégré à la membrane durant la synthèse.

Les protéines intégrées dans le REG peuvent être l'objet de modifications co- et post-traductionnelles comme par exemple :

- glycosylation cotraductionnelle sur le groupement amine des résidus asparagine de la protéine.
- établissement de ponts disulfures entre cystéines (post-traductionnel).

Les produits de la synthèse du REG sont ensuite transportés vers l'appareil de Golgi. On distingue généralement un compartiment intermédiaire entre REG et Golgi, baptisé ERGIC (pour ER to Golgi Intermediate Compartment). Ce compartiment permet le tri des protéines quittant le REG en rendant possible le retour des protéines résidentes du REG. Ces protéines ont en commun de porter, à leur extrémité C-terminale, une séquence particulière, appelée séquence KDEL, en raison des 4 résidus présents sur l'extrémité C-terminale et responsables de la rétention de la protéine dans le réticulum.

1.2.2.2 l'appareil de Golgi

Rappelons que le Golgi est avant tout un compartiment de **transit** où les protéines peuvent être l'objet de glycosylations successives, avant leur sécrétion (export)

ou adressage vers un lysosome.

Ainsi, l'ERGIC délivre les protéines au cis-Golgi, où elles passent progressivement du cis-Golgi vers le Golgi médian, puis du Golgi médian vers le trans-Golgi, pour finalement atteindre le trans-Golgi network (TGN) (figure 1.5). Les mécanismes du transport à travers les différentes parties du Golgi sont complexes et ne seront pas détaillés ici.

1.2.2.3 les lysosomes

Les lysosomes représentent un compartiment acide d'accumulation d'enzymes hydrolytiques. Si une partie des protéines constitutives des lysosomes proviennent du TGN, elles peuvent aussi provenir de la fusion avec d'autres structures vésiculaires (les mécanismes de création sont montrés à titre indicatif sur la figure 1.5).

1.2.3 Mitochondries, chloroplastes, péroxysomes

Ces organites sont liés au métabolisme oxydatif des cellules. Si les mitochondries et les chloroplastes ont des rôles de "transformateurs énergétiques" cellulaire, les péroxisomes correspondent à des compartiments d'oxydation de métabolites sans récupération d'énergie. Chez l'homme les péroxisomes se trouvent en forte densité dans les cellules du foie.

Bien que les mitochondries et chloroplastes possèdent leur propre génome, résidu ancestral de leur origine bactérienne, et permettant la synthèse de protéines à l'intérieur de l'organite, la grande majorité des protéines localisées dans ces compartiments sont d'origine nucléaire. On estime que seulement 5 à 10% des protéines des mitochondries sont d'origine mitochondriale.

L'adressage de ces protéines à une mitochondrie ou à un chloroplaste est généralement effectué par la reconnaissance post-traductionnelle d'un signal présent sur leur extrémité N-terminale.

Deux classes de signaux ont été identifiées pour l'adressage des protéines au péroxisome :

- Un signal C-terminal reconnu dans le cytoplasme par la protéine Pex5, généralement caractérisé par un tripeptide (PTS1).
- Un signal de neuf résidus situé près de l'extrémité N-terminale de la protéine (PTS2).

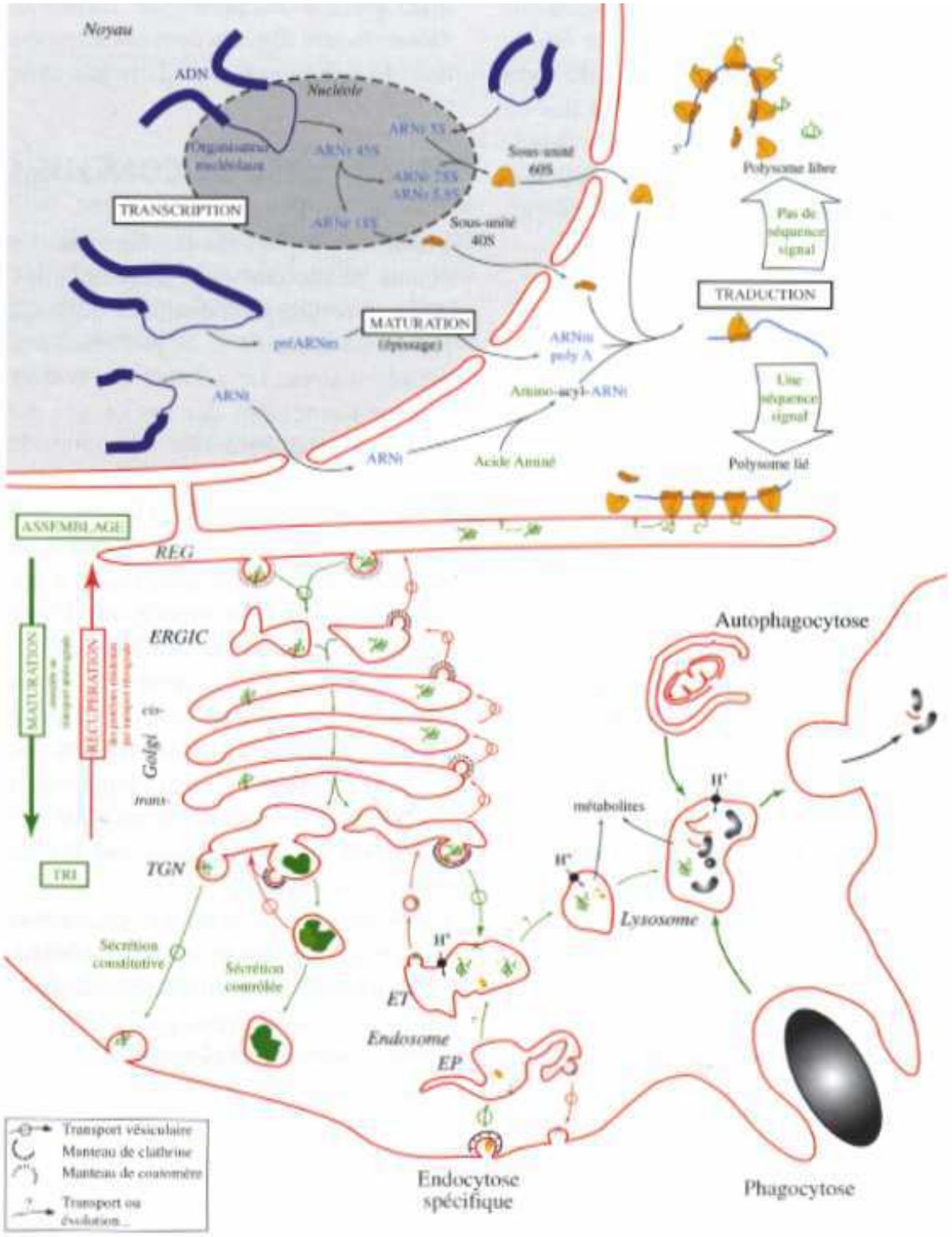


FIG. 1.5 – Schéma récapitulatif des traffics cellulaires, tiré du livre de Y. Bassaglia, “Biologie Cellulaire”. Les acides nucléiques sont transcrits et subissent une maturation dans le noyau ; leur adressage ultérieur dépend des protéines à la synthèse desquelles ils participent. Les protéines sont aiguillées vers les différentes voies à l’occasion de carrefours importants : dans les endosomes précoces pour les protéines endocytées (non abordé dans le texte), dès la traduction et au niveau du TGN pour les protéines synthétisées par la cellule. Pour ces protéines, le REG apparaît comme un lieu d’assemblage, le Golgi comme une zone de maturation, dans laquelle des processus de récupération rétrograde permettent la rétention de protéines résidentes avant la zone de tri du TGN (les mécanismes d’import dans le péroxisome, la mitochondrie et le chloroplaste ne sont pas indiqués).

1.3 Détermination expérimentale

Une méthode courante de détermination de la localisation intracellulaire d'une protéine utilise le marquage à la protéine fluorescente. Pour ce faire, le gène d'intérêt est fusionné avec un rapporteur fluorescent (GFP pour "Green Fluorescent Protein") à l'une de ses deux extrémités (qui correspondront donc aux extrémités C et N-terminale de la protéine traduite). La localisation de la protéine est ensuite observée au microscope optique par fluorescence.

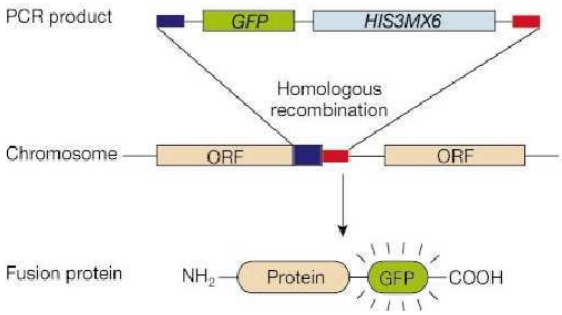


FIG. 1.6 – Schéma résumant le procédé de marquage par protéine de fluorescence. Pour le gène d'intérêt, la séquence nucléique correspondant à la protéine de fluorescence est intégrée à la place du codon stop par recombinaison homologue. La protéine chimérique produite émet donc de la fluorescence.

Cette méthode est cependant limitée à cause de la faible résolution de la microscopie optique (0,2 μm) et par les artefacts de localisation que peut entraîner la fusion au rapporteur (par exemple, la GFP fusionnée peut masquer le signal de rétention dans le réticulum endoplasmique ou le signal d'adressage au péroxisome PST1). Dans le cas de la levure, ces méthodes ont été développées à grande échelle, en fusionnant une séquence codante différente par souche.

Deuxième partie

Modèles markoviens et classification

Chapitre 2

Modèles markoviens et séquences biologiques

Comme abordé au chapitre précédent, les molécules biologiques sont constitués d'une chaîne carbonée ou phosphatée orientée à laquelle un nombre fini de radicaux peuvent être accrochés (4 dans le cas de l'ADN ou l'ARN et 20 à 22 dans le cas des séquences peptidiques). Une première approximation unidimensionnelle d'une séquence biologique peut donc être une chaîne de caractères énumérant la suite des susdits radicaux.

Mathématiquement, on prend donc le parti de travailler sur des processus $\{Y_t\}_{t\geq 1}$ à valeur dans Σ, ensemble de cardinal fini. Afin de prendre en compte la composition des séquences biologiques, nous présentons dans ce chapitre les modèles de type markovien.

Le modèle de chaîne de Markov (CM) d'ordre k décrit la probabilité d'apparition d'un état, ici une lettre, conditionellement aux précédents comme ne dépendant que de l'information des k derniers. Plus formellement, la loi d'apparition de la lettre à la t$^{\text{ème}}$ position s'écrit :

$$\mathbf{P}(Y_{t+1} = y_{t+1} | Y_1^t = y_1^t) = \mathbf{P}(Y_{t+1} = y_{t+1} | Y_{t-k+1}^t = y_{t-k+1}^t) = \Pi(y_{t-k+1}^t, y_{t+1})$$

où Y_t^u correspond au vecteur des variables aléatoires $(Y_t, Y_{t+1}, \ldots, Y_u)^\top$ et y_t^u à la séquence des observés.

Ainsi, dans le cas des chaînes de Markov d'ordre k, la loi d'apparition d'une lettre à une position donnée ne dépend que des k lettres précédentes. On parlera alors du contexte d'une lettre.

Cependant, ce modèle souffre de plusieurs limitations :

- la composition est supposée homogène le long de la séquence.
- un ordre de lecture est imposé sur la séquence, par exemple la séquence d'ADN est nécessairement lue de 5' vers 3'.

– Un modèle estimé n'apporte pas *per se* d'information sur une séquence, si ce n'est dans une approche comparative.

Les modèles de Markov cachés repoussent une partie de ces limitations en permettant l'aternance de plusieurs chaînes de Markov le long de la séquence. Leur succession est elle aussi contrôlée par une chaîne de Markov. Ainsi, on suppose qu'au processus de génération des lettres $(Y_t)_{t=1}^n$ est couplé un processus caché $(S_t)_{t=1}^n$ déterminant la loi d'apparition des observations. La suite S_t, à valeurs dans l'espace fini $\mathcal{S}$, conditionne la chaîne de Markov utilisée pour l'apparition des lettres. De plus, les états S_t se succèdent suivant une chaîne de Markov (généralement d'ordre 1).

Dans cette approche, la segmentation de la séquence suivant la suite des états cachés apporte alors une annotation sur la séquence.

Plus formellement, les relations permettant de définir une Chaîne de Markov Cachée (CMC) s'écrivent ainsi :

$$\mathbf{P}(S_{t+1} = s_{t+1} \mid S_t = s_t, \dots S_1 = s_1) = \mathbf{P}(S_{t+1} = s_{t+1} \mid S_t = s_t) \quad \text{(CM sur les états)} \quad (2.1)$$

$$\mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_1^{t-1} = y_1^{t-1}, S_1^{t+1} = s_1^{t+1}) = \mathbf{P}(Y_{t+1}, y_{t+1} \mid Y_{t-1} = y_{t-1}, S_{t+1} = s_{t+1})$$
$$\text{(CM conditionellement à l'état)} \quad (2.2)$$

Les sections 2.1 et 2.2 présentent les propriétés mathématiques des chaînes de Markov et des chaînes de Markov cachées, leur méthode d'estimation, ainsi que quelques applications de ces modèles en bioinformatique (2.1.3).

2.1 Chaînes de Markov

2.1.1 Définition et premières propriétés

Cette section présente les chaînes de Markov, leurs propriétés élémentaires et un résultat sur le comportement asymptotique de la chaîne. Les chaînes de Markov ont donné lieu à de multiples ouvrages dont le caractère généraliste dépasse de loin notre cadre d'étude. Le lecteur intéressé pourra se référer aux ouvrages de Karlin [KH66] ou Freedman [Fre71] pour une présentation plus générale.

2.1.1.1 Définitions

Définition 2.1. Soit Σ un ensemble de cardinal fini, et $\{Y_t\}_{t\geq 1}$ un processus à valeurs dans Σ. Soit μ_0 la loi de génération de la première lettre : $\mathbf{P}(Y_1 = a) = \mu_0(a)$, $\forall\, a \in \Sigma$

Alors $\{Y_t\}_{t\geq 1}$ est appelé une **chaîne de Markov (CM)** d'ordre 1 si et seulement si, pour tout $t > 1$ et $y_1, \dots, y_t, y_{t+1} \in \Sigma$:

$$\mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_t = y_t, Y_{t-1} = y_{t-1}, \dots, Y_1 = y_1) = \mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_t = y_t) \quad (2.3)$$

Les chaînes de Markov sont donc des processus à mémoire finie : la loi de la t$^{\text{ième}}$ variable aléatoire ne dépend que de la valeur prise par la variable précédente. C'est aussi l'extension la plus logique mathématiquement à la loi multinomiale (pouvant ainsi être vue comme une chaîne de Markov d'ordre 0).

Ajoutons à l'équation (2.3) l'hypothèse de stationnarité de la chaîne. Si $\{Y_t\}_{t\geq 1}$ est une CM, on suppose que les probabilités

$$p_{a\,|\,b} = \mathbf{P}(Y_{t+1} = b \,|\, Y_t = a), \quad a, b \in \Sigma, t \geq 1$$

sont indépendantes de t. Une telle CM est dite *homogène*, et peut alors être résumée par une matrice Π de dimension $|\Sigma| \times |\Sigma|$:

$$\begin{aligned} \Pi :& \Sigma \times \Sigma \longrightarrow [0,1] \\ & (a,b) \longmapsto \Pi(a,b) = \mathbf{P}(Y_{t+1} = b \,|\, Y_t = a), t > 1 \end{aligned}$$

avec la contrainte (stochasticité de la matrice) :

$$\forall\, a \in \Sigma, \sum_{b\in\Sigma} \Pi(a,b) = 1$$

On appelle Π la *matrice de transition* de Y_t.

Insistons sur le fait qu'une CM homogène est entièrement décrite par sa matrice de transition associée Π, où les lignes correspondent aux événements conditionnant et les colonnes à l'événement observé. $\Pi(T, A)$ correspond à la probabilité d'observer A sachant que l'on vient d'observer T. On voit aussi que le nombre de paramètres linéairement indépendants permettant de définir une CM est $|\Sigma|(|\Sigma| - 1)$. Un autre avantage de la représentation matricielle est de permettre la caractérisation algébrique des propriétés de la CM.

Par exemple, on vérifie aisément que la loi de succession entre les lettres distantes de n positions est une CM de matrice de transition Π^n. Dans le cas $n = 2$, on a :

$$\begin{aligned} \mathbf{P}(X_{t+2} = b \,|\, X_t = a) &= \sum_{u\in\Sigma} \mathbf{P}(X_{t+2} = b \,|\, X_t = a, X_{t+1} = u) \\ &= \sum_{u\in\Sigma} \mathbf{P}(X_{t+2} = b \,|\, X_{t+1} = u) \cdot \mathbf{P}(X_{t+1} = u \,|\, X_t = a) \\ &= \sum_{u\in\Sigma} \Pi(a,u) \cdot \Pi(u,b) = \Pi^2(a,b) \end{aligned}$$

le cas général se démontre aisément par récurrence.

On peut étendre la définition 2.1 pour des processus possédant une mémoire supérieure à une lettre. On parlera alors de chaîne de Markov d'ordre k si la loi d'apparition d'une lettre est conditionnée par le contexte des k précédentes lettres (on parle aussi de processus de mémoire k).

Définition 2.2. Un processus Y_t à valeurs dans Σ ($t = 1, \ldots, l$), de loi initiale μ_0 sur Y_1^k (*i.e.* $\mathbf{P}(Y_1^k = y_1^k) = \mu_0(y_1^k)$) est appelé une **chaîne de Markov d'ordre** k si et seulement si, pour tout $t > k$:

$$\begin{aligned}\mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_t = y_t, Y_{t-1} = y_{t-1}, \ldots, Y_1 = y_1) \\ &= \mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_t = y_t, Y_{t-1} = y_{t-1}, Y_{t-k+1} = y_{t-k+1}) \\ &:= \Pi(y_{t-k+1} \ldots y_t\,, y_{t+1})\end{aligned}$$

La matrice de transition Π du processus s'écrit alors :

$$\begin{aligned}\Pi : \Sigma^k \times \Sigma &\longrightarrow [0,1] \\ (a_1 \ldots a_k, b) &\longmapsto \Pi(a_1 \ldots a_k, b)\end{aligned}$$
$$\forall\, (a_1 \ldots a_k) \in \Sigma^k, \sum_{b \in \Sigma} \Pi(a_1 \ldots a_k, b) = 1$$

Remarquons une première propriété des modèles de mémoire k. Par un artifice de changement d'alphabet, toute chaîne d'ordre k peut se réécrire comme une chaîne d'ordre 1. La construction de cet alphabet est résumée dans la proposition suivante.

Proposition 2.3. *Soit* (Y_t, Π, Σ) *une chaîne de Markov d'ordre k. Alors, notant* $\tilde{\Sigma}$ *l'alphabet défini sur les mots de k lettres :* $\tilde{\Sigma} = \left\{ a_1 a_2 \ldots a_k ; a_i \in \Sigma, 1 \leq i \leq k, \right\}$. *Il existe une matrice* $\tilde{\Pi}$ *définie sur* $\tilde{\Sigma}$ *telle que :*

$$\tilde{\Pi}(a_1 \ldots a_k, b_1 \ldots b_k) = \begin{cases} \Pi(a_1 \ldots a_k, b) & \text{si } a_{i+1} = b_i,\ \forall\, i < k \\ 0 & \text{sinon} \end{cases}$$

Toute chaîne de Markov d'ordre k se réécrit donc comme une chaîne de Markov d'ordre 1 à valeurs dans l'alphabet des mots de k lettres (ou k-mers) chevauchants.

Ce résultat se comprend aisément. En guise d'exemple, prenons le cas d'une chaîne de Markov d'ordre 2 et de matrice Π sur l'alphabet nucléotidique. La matrice de transition $\tilde{\Pi}$ se réécrit alors comme montré dans la figure 2.1. Le nombre de paramètres décrivant un modèle d'ordre k est $|\Sigma|^k(|\Sigma| - 1)$. Notons que cette augmentation exponentielle du nombre de paramètres avec l'ordre de la CM peut, à l'estimation, poser certains problèmes. Le nombre de paramètres à estimer peut ainsi devenir trop grand en regard du nombre d'observés.

Dans la suite de ce chapitre, en vertu de la proposition 2.3 et afin d'alléger les notations, on ne considérera que des CM d'ordre 1, en gardant à l'esprit les contraintes calculatoires dues à l'augmentation exponentielle du nombre de paramètres avec la mémoire du processus.

$$\Pi = \begin{pmatrix}
 & aa & \dots & at & ca & \cdots & ct & ga & \cdots & gt & ta & \cdots & tt \\
aa & \pi(aa,a) & \cdots & \pi(aa,t) & 0 & \dots & & & & & & & 0 \\
ac & 0 & \cdots & 0 & \pi(ac,a) & \cdots & \pi(ac,t) & 0 & \dots & & & & 0 \\
\vdots & \dots & & & & & & \ddots & \dots & & & & \\
ta & \pi(ta,a) & \cdots & \pi(ta,t) & 0 & \dots & & & & & & & 0 \\
tc & 0 & \cdots & 0 & \pi(tc,a) & \cdots & \pi(tc,t) & 0 & \dots & & & & 0 \\
tg & 0 & \dots & & & & 0 & \pi(tg,a) & \cdots & \pi(tg,t) & 0 & \cdots & 0 \\
tt & 0 & \dots & & & & & & & 0 & \pi(tt,a) & \cdots & \pi(tt,t)
\end{pmatrix}$$

FIG. 2.1 – Matrice de transition pour une chaîne de Markov d'ordre 2 dans l'aphabet augmenté des dinucléotides

2.1.1.2 Classification des états d'une CM

Remarquons dans un premier temps l'équivalence entre la matrice de transition d'une CM et sa représentation par un graphe orienté. Alors, une suite d'observés (y_1^t) correspond exactement à une chaîne dans le graphe, et la probabilité de la séquence $y_1 \dots y_t$ se calcule directement par multiplication sur les poids associées aux arcs empruntés.

$$\Pi = \begin{pmatrix}
 & \text{A} & \text{C} & \text{G} & \text{T} \\
\text{A} & 0.2 & 0.1 & 0.4 & 0.3 \\
\text{C} & 0.1 & 0.4 & 0.3 & 0.2 \\
\text{G} & 0 & 0 & 0 & 1 \\
\text{T} & 0.5 & 0.5 & 0 & 0
\end{pmatrix}$$

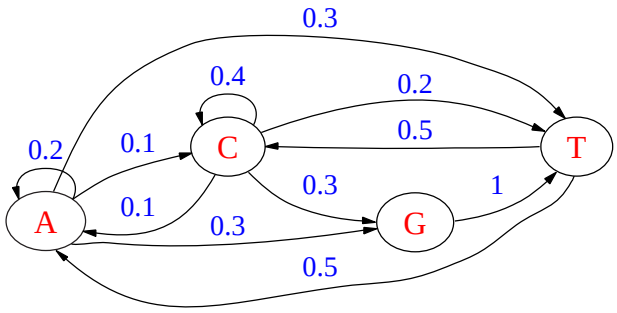


FIG. 2.2 – Relation entre une chaîne de Markov et le graphe correspondant pour une CM d'ordre 1 sur $\Sigma = \{A, C, G, T\}$.

Cette représentation met en avant les propriétés de certains états ou groupes d'états. Par exemple, on voit dans la figure 2.2, qu'un G ne peut être suivi que par un T

Définition 2.4. Considérons deux états a et b de Σ et une CM Π.

– On dit que b est atteint de a et on note $a \rightsquigarrow b$ si il existe $n > 0$ tel que $\Pi^n(a,b) > 0$

– Si $a \rightsquigarrow b$ et $b \rightsquigarrow a$, on dit que a et b *communiquent* et on note $a \leftrightsquigarrow b$.

On vérifie aisément que "$\leftrightsquigarrow$" définit une relation d'équivalence sur Σ. Les classes d'équivalence ainsi définies – qui correspondent aux composantes connexes du graphe associé – définissent donc une partition des états de la chaîne.

Cette relation est aussi liée aux sous ensembles dits *absorbants* de la chaîne de Markov. Si X_1 appartient à $\mathcal{A}$ absorbant, il ne sortira jamais de celui-ci. On peut aussi définir les états absorbants de manière équivalente par les parties $\mathcal{A} = (a_1, \ldots, a_t)$ de Σ telles que :

$$\forall\, b \in \Sigma - \mathcal{A}, \forall\, n > 0, \qquad \Pi^n(a_i, b) = 0; i = 1, \ldots, t$$

Si la chaîne possède comme seules classes d'équivalence $\varnothing$ et Σ, on dit que celle ci est irréductible. Autrement dit, si $\forall\, a, b \in \Sigma, a \leftrightsquigarrow b$. Si la chaîne n'est pas irréductible, on peut aisémement se ramener à une chaîne irréductible sur chaque classe d'équivalence C en imposant une loi initiale ne chargeant que les états dans C.

Dans le cas des séquences de caractères, nous verrons par la suite que la forme de l'estimateur du maximum de vraisemblance nous amène à travailler uniquement avec des chaînes irréductibles.

Définition 2.5 (périodicité). Si $a \rightsquigarrow a$, alors la période de a (notée $\tau(a)$) est définie comme :

$$\tau(a) = \text{PGCD}\{n \in \mathbb{N}\,;\, \Pi^n(a, a) > 0\}$$

Si tous les états d'une CM sont de période 1, on dit que la chaîne est apériodique.

Afin d'illustrer un exemple de chaîne périodique, introduisons une extension des modèles de Markov utilisé pour l'analyse de l'ADN des séquences codant pour des gènes : les chaînes de Markov phasées. Pour tenir compte de la structure en codons des gènes, on suppose disposer de trois matrices de transition (Π_1, Π_2, Π_3) gouvernant l'apparition des lettres à chaque phase du cadre de lecture.

$$\mathbf{P}(X_t = x_t | X_1 = x_1, \ldots, X_n = x_n) = \Pi_{t\%3}(x_{t-1}, x_t)$$

où "$n\%d$" représente le reste de la division euclidienne de n par d.

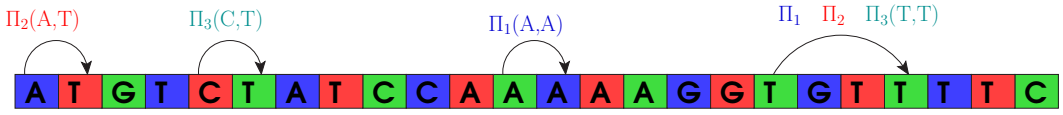


FIG. 2.3 – réalisation d'un modèle de Markov phasé sur une séquence.

Ce processus peut se réécrire comme une chaîne de Markov d'ordre 1 sur l'alphabet à $3 \times \Sigma$ lettres, $\tilde{\Sigma} = \left\{(a, m), a \in \Sigma, m = 1, 2, 3\right\}$ avec comme matrice de transition :

$$\tilde{\Pi}\Big((a, p)\,,\,(b, q)\Big) = \begin{cases} \Pi_q(a, b) & \text{si } p + 1 = q \\ 0 & \text{sinon} \end{cases} \qquad \text{ou} \qquad \tilde{\Pi} = \begin{pmatrix} 0 & \Pi_2 & 0 \\ 0 & 0 & \Pi_3 \\ \Pi_1 & 0 & 0 \end{pmatrix}$$

La structure creuse de $\tilde{\Pi}$ permet de calculer rapidement ses puissances :

$$\tilde{\Pi}^2 = \begin{pmatrix} 0 & 0 & \Pi_2\Pi_3 \\ \Pi_1\Pi_3 & 0 & 0 \\ 0 & \Pi_1\Pi_2 & 0 \end{pmatrix}, \quad \tilde{\Pi}^3 = \begin{pmatrix} \Pi_2\Pi_3\Pi_1 & 0 & 0 \\ 0 & \Pi_3\Pi_1\Pi_2 & 0 \\ 0 & 0 & \Pi_1\Pi_2\Pi_3 \end{pmatrix}, \quad \tilde{\Pi}^4 = \begin{pmatrix} 0 & \bullet & 0 \\ 0 & 0 & \bullet \\ \bullet & 0 & 0 \end{pmatrix}$$

La structure périodique des puissances de $\tilde{\Pi}$ montre que le temps de retour pour n'importe quelle lettre est nécessairement un multiple de 3. Donc $\tau(a, i) = 3$ pour $(a, i) \in \tilde{\Sigma}$.

Comme nous le verrons dans le paragraphe suivant, les propriétés d'irréductibilité et d'apériodicité d'une chaîne de Markov permettent de montrer la convergence de celle ci vers une mesure stationnaire, indépendante de la distribution initiale.

2.1.1.3 Convergence des chaînes de Markov vers la loi stationnaire.

Cette section aborde l'étude du comportement asymptotique d'une chaîne de Markov, *i.e.* à quelles conditions il existe une unique mesure μ sur Σ telle que, quelle que soit la loi initiale μ_0, $\lim_{t\to\infty} \Pi^t = \mu$. La connaissance de la loi stationnaire permet alors de calculer la probabilité pour une suite de caractères d'apparaître à une position donnée, la chaîne étant supposée à l'équilibre.

Les propriétés de convergence d'une CM peuvent être établies en utilisant la classification des états introduite en section précédente et le théorème du renouvellement de Feller ([Fre71]). Dans la suite, nous nous attacherons à donner simplement une justification algébrique, utilisant les propriétés de la matrice de transition.

La preuve de convergence s'appuie sur le théorème de Perron-Frobénius, que nous énonçons simplement ici. Le lecteur intéressé pourra se référer à ([KH66]) pour plus de détails.

Théorème 2.6 (Perron-Frobénius). *Soit M une matrice $n \times n$ dont toutes les composantes sont positives ou nulles et telle que M^n a toutes ses composantes strictement positives pour un $n \in \mathbb{N}$. Alors :*

(i) M admet une valeur propre positive λ_0 dont un vecteur propre associé v_0 a toutes ses composantes strictement positives.

(ii) λ_0 est de multiplicité 1.

(iii) Si λ est une autre valeur propre de M, $|\lambda| < \lambda_0$

Remarquons d'abord que toute matrice de transition possède la valeur propre 1. En effet, comme les lignes de la matrice somment à 1, on a, pour tout vecteur $\mu = (c, c, \ldots, c)^\top$, $\Pi\mu = \mu$.

Montrons maintenant que 1 est la seule valeur propre pouvant être associée à un vecteur propre à composantes positives pour une matrice stochastique. Soit v un vecteur propre à gauche de Π , à composantes toutes positives et associé à une valeur propre λ.

On a alors $\lambda v_i = \sum_{j=1}^{|\Sigma|} v_j \Pi(j,i)$ pour i dans Σ. En sommant sur les lignes de la matrice, on obtient $\lambda \sum_i v_i = \sum_j v_j$ (la matrice est stochastique). Comme les v_j sont positifs ou nuls on voit qu'on a nécessairement $\lambda = 1$.

Théorème 2.7 (Convergence des chaînes de Markov). *Soit une chaîne de Markov de matrice de transition Π à valeurs sur Σ, irréductible et apériodique. Alors Π^n converge vers une unique loi stationnaire μ, indépendante de la distribution initiale. Autrement dit :*

$$\exists \mu \; tq \; \forall \, a, b \in \Sigma, \lim_{n\to\infty} \Pi^n(a,b) = \mu(b)$$

De plus, notant la loi stationnaire μ par le vecteur ligne $\big(\mu(a)\big)_{a\in\Sigma}$, elle vérifie :

$$\mu\Pi = \mu$$

Démonstration. Remarquons dans un premier temps que la matrice de transition d'une chaîne de Markov irréductible apériodique vérifie les conditions permettant d'appliquer le théorème de Perron Frobénius, *i.e.* il existe m_0 tel que Π^{m_0} est à composantes strictement positives.

Notons $\mathbf{u}_i$ le vecteur propre à droite associé à la valeur propre λ_i. Par commodité, on range les λ_i en ordre décroissant (*e.g.* $\lambda_1 = 1$). Alors, si on note L la matrice des $\mathbf{u}_i$ rangés en ligne, par irréductibilité de la chaîne, L est inversible et on a : $\Pi = L^{-1}\Delta L$, avec Δ=`diag`$(\lambda_1, \ldots, \lambda_n)$. En outre, les colonnes de L^{-1} correspondent aux vecteurs propres à gauche de Π qu'on notera $\mathbf{v}_i$. On peut réécrire la relation précédente comme :

$$\Pi = \sum_{i=1}^{|\Sigma|} \lambda_i \mathbf{v}_i' \mathbf{u}_i = \sum_{i=1}^{|\Sigma|} \lambda_i P_i \quad \text{et} \quad \Pi^n = \sum_{i=1}^{|\Sigma|} \lambda_i^n P_i$$

avec en particulier :

$$P_1 = \begin{pmatrix} \mu_1 & \mu_2 & \ldots & \mu_{|\Sigma|} \\ \vdots & \vdots & \ldots & \vdots \\ \mu_1 & \mu_2 & \ldots & \mu_{|\Sigma|} \end{pmatrix} \quad \text{où} \quad (\mu_1, \ldots, \mu_{|\Sigma|}) \cdot \Pi = \mu$$

Soit μ_0 un vecteur initial quelconque, on a

$$\mu_0\Pi^n = \mu_0 P_1 + \sum_{i=2}^{|\Sigma|} \lambda_i^n \mu_0 P_i \quad \text{et donc} \quad \lim_{n\to\infty} \mu_0\Pi^n = \mu_0 P_1 = \mu$$

□

Pour des raisons pratiques, notamment pour le calcul de la distance en variation totale entre deux CM, on considère couramment que la distribution initiale de la chaîne de Markov est sa distribution stationnaire.

Remarque 2.8. L'hypothèse d'apériodicité est essentielle pour l'existence d'une mesure invariante. En effet, reprenons l'exemple présenté dans la section précédente sur les modèles de Markov phasés. On voit que la forme de Π^n n'est pas stable suivant la valeur de $n\%3$. Cependant, on constate aisément que la suite des Π^{3n} converge vers une mesure stationnaire $\tilde{\mu} = (\mu_1, \mu_2, \mu_3)$. Les μ_i correspondent aux lois stationnaires à phase fixée et vérifient la relation :

$$\mu_i \cdot \Pi_{(i+1)\%3}\, \Pi_{(i+2)\%3}\, \Pi_i = \mu_i$$

Plus généralement, on peut montrer que pour une CM Π irréductible et périodique de période d, Π^{nd} converge vers une unique mesure stationnaire quand $n \to \infty$.

Rappelons la définition de la distance en variation totale dans le cadre d'ensembles finis et pour des chaînes de Markov.

Définition 2.9 (Distance en variation totale). Soient μ_1 et μ_2 deux mesures sur un ensemble fini A. La distance en variation totale entre μ_1 et μ_2 notée $d_{VT}(\mu_1, \mu_2)$ est définie par :

$$d_{VT}(\mu_1, \mu_2) = \sum_{a \in A} \left|\mu_1(a) - \mu_2(a)\right|$$

Cette distance correspondant à la norme l^1 et peut donc être notée de manière équivalente : $\left\|\mu_1 - \mu_2\right\|_1$

Soient $(X_1, \ldots, X_l)$, $(Y_1, \ldots, Y_l)$ deux chaînes de Markov d'ordre 1 à valeur sur Σ, de matrices de transition respectives Π_1 et Π_2, et de lois stationnaires μ_1 et μ_2. Supposons que la chaîne est en régime stationnaire (*i.e.* $\mathcal{L}(X_1) \sim \mu_1$ et $\mathcal{L}(Y_1) \sim \mu_2$). La distance en variation totale entre $(X_1, \ldots, X_l)$ et $(Y_1, \ldots, Y_l)$ se note comme la distance en variation totale entre les deux matrices de transition Π_1 et Π_2 :

$$d_{VT}(\Pi_1, \Pi_2) = \sum_{a \in \Sigma} \sum_{b \in \Sigma} |\mu_1(a)\Pi_1(a, b) - \mu_2(a)\Pi_2(a, b)|$$

Notons que l'hypothèse de stationnarité est essentielle pour la simplification du calcul dans le cas des chaînes de Markov. Si cette hypothèse n'était plus faite, il serait nécessaire de fixer l'espace des états aux séquences de longueur fixée.

Théorème 2.10 (vitesse de convergence vers la loi stationnaire). *Soit une chaîne de Markov de matrice de transition Π à valeurs sur Σ, irréductible et apériodique. Notant λ_i ($i = 1, \ldots, |\Sigma|$), les valeurs propres de Π rangées en ordre décroissant, et μ la mesure stationnaire, $\exists C$ tel que :*

$$\forall\, n \in \mathbb{N},\ \max_{a \in \Sigma} \left(\mathrm{d}_{VT}(\Pi^n(a, \bullet), \mu)\right) \leq C.\,|\lambda_2|^n$$

Démonstration. Supposons la chaîne générée à partir d'une distribution $\mathbf{x}_0$ quelconque, on a :

$$\mathbf{x}_0\Pi^k = \sum_{i=1}^{|\Sigma|} \lambda_i^k \mathbf{x}_0 \mathbf{v}_i' \mathbf{u}_i$$

Or, on a $\mathbf{v}_1'\mathbf{u}_1 = \begin{pmatrix} \mu_1 & \mu_2 & \dots & \mu_{|\Sigma|} \\ \vdots & \vdots & \dots & \vdots \\ \mu_1 & \mu_2 & \dots & \mu_{|\Sigma|} \end{pmatrix}$, et on peut donc réécrire $\mathbf{x}_0\Pi^k$ comme : $\mathbf{x}_0\Pi^k = \mu + \sum_{i=2}^{|\Sigma|} \lambda_i^k \mathbf{x}_0 \mathbf{v}_i' \mathbf{u}_i$

En passant μ dans le membre de droite, et en appliquant la norme :

$$\left\|\mathbf{x}_0\Pi^k - \mu\right\| \leq \sum_{i=2}^{|\Sigma|} |\lambda_i|^k \left\|\mathbf{x}_0 \mathbf{v}_i' \mathbf{u}_i\right\|$$

$$\left\|\mathbf{x}_0\Pi^k - \mu\right\| \leq |\lambda_2|^k \underbrace{(|\Sigma| - 1) \max_{i=2,\dots,|\Sigma|} \left\|\mathbf{x}_0 \mathbf{v}_i' \mathbf{u}_i\right\|}_{C}$$

l'inégalité existant pour tout $\mathbf{x}_0$, le point est démontré. □

Ce résultat fait apparaître que l'influence de la loi initiale décroit exponentiellement avec la longueur de la séquence. On parle généralement de la propriété de "mémoire courte" des chaînes de Markov. Cette propriété peut être utilisée pour certains calculs numériques faisant intervenir des conditionnements successifs dans la complexité générale. Nous verrons un exemple simple dans la section 2.1.3.

La distance "d'oubli" du conditionnement dépend, à une constante près, de la seconde plus grande valeur propre de la matrice de transition. Le stage de David Gomes, effectué sous la direction de Grégory Nuel dans le laboratoire, a visé à étudier les valeurs de ces secondes valeurs propres dans le cas où la chaîne de Markov est estimée sur une séquence biologique. L'analyse a porté sur des génomes complets aussi bien que sur des régions regroupées par catégorie fonctionnelle. (séquences codantes, régions régulatrices, intergénique...). Les résultats semblent indiquer que les secondes valeurs propres restent stables quel que soit le type de séquence étudiée. La distance en variation totale entre l'itéré de la matrice de transition et la loi stationnaire deviendrait donc largement inférieure à la précision machine dès une quarantaine de lettres. Cette constatation justifie dans une certaine mesure l'hypothèse de stationnarité pour la loi initiale de la CM, les longueurs des séquences étudiées étant en général de plusieurs ordres de grandeur supérieures à la distance de convergence de la chaîne.

2.1.2 Estimation

Cette section rappelle les méthodes d'estimation pour une CM ainsi que les propriétés asymptotiques des estimateurs. Dans la suite, $\boldsymbol{\theta}$ fera couramment référence au

vecteur des paramètres du modèle. Plus précisement, on notera $\boldsymbol{\theta}^*$ le vecteur des paramètres ayant permis la génération des séquences et $\hat{\boldsymbol{\theta}}$ le vecteur des paramètres estimés par maximum de vraisemblance. Rappelons que le nombre de paramètres linéairement indépendants dans une CM d'ordre 1 vaut $|\Sigma|\cdot(|\Sigma|-1)$ (et $|\Sigma|^k\cdot(|\Sigma|-1)$ pour une CM d'ordre k). Aussi, $\boldsymbol{\theta}$ pourrait directement être identifié à Π en retirant de cette dernière une colonne. Ainsi, dans la suite, $\boldsymbol{\Pi}$ fera implicitement référence aux vecteur de paramètres $\boldsymbol{\theta}$ et correspondra donc, sauf indication contraire à Π auquel on aura retiré une colonne.

2.1.2.1 Estimation par maximum de vraisemblance

Commençons par quelques notations, soit une séquence observée de longueur ℓ, $\mathbf{S} = y_1^\ell$. On note :

- $N_{\mathbf{S}}(a_1 \dots a_k)$ le nombre d'occurrence du mot $a_1 \dots a_k$ dans la séquence $\mathbf{S}$
- $N_{\mathbf{S}}(a_1 \dots a_k\bullet) = \sum_{b\in\Sigma} N_{\mathbf{S}}(a_1 \dots a_k b)$ le nombre d'occurence dont $a_1 \dots a_k$ est préfixe dans $\mathbf{S}$.

La vraisemblance de $\mathbf{S}$ sous un modèle M_1 de matrice de transition Π et de loi initiale μ_0 s'écrit :

$$
\begin{aligned}
L(\mathbf{S},\Pi) &= \mu_0(y_1)\cdot\prod_{i=2}^{\ell}\Pi(x_{i-1},x_i)\\
\log L(\mathbf{S},\Pi) = \mathcal{L}(\mathbf{S},\Pi) &= \log(\mu_0(y_1)) + \sum_{i=2}^{\ell}\log\Pi(x_{i-1},x_i)\\
&= \log(\mu_0(y_1)) + \sum_{i=2}^{\ell}\sum_{(a,b)\,in\,\Sigma^2}\mathbb{I}_{\{x_{i-1}=a\}}.\mathbb{I}_{\{x_i=b\}}.\log\Pi(x_{i-1},x_i)\\
&= \log(\mu_0(y_1)) + \sum_{(a,b)\,in\,\Sigma^2} N_{\mathbf{S}}(ab)\log\Pi(a,b)
\end{aligned}
$$

Donc, pour estimer $\hat{\Pi}$ sur $\mathbf{S}$ par maximum de vraisemblance, on peut se ramener au problème d'optimisation suivant sur les α_{ab} :

$$
\begin{aligned}
&\min_{\{\alpha_{ab},\,a,b\in\Sigma\}} -C - \sum_{(a,b)\in\Sigma^2} N_{\mathbf{S}}(a,b)\log(\alpha_{ab})\\
&\text{soumis à : } \forall\, a\in\Sigma, \sum_{b\in\Sigma}\alpha_{ab} = 1,\\
&\qquad\qquad \forall a,b\in\Sigma, \alpha_{ab} > 0
\end{aligned}
$$

La fonction est convexe et bien définie, il existe donc une unique solution. Ecrivons le Lagrangien associé au problème (pour plus de détails, on pourra se reporter à l'annexe

ou à la section 3.3.2) :

$$\mathrm{L}(\boldsymbol{\alpha},\boldsymbol{\beta}) = C + \sum_{(a,b)\in\Sigma^2} N_{\mathbf{S}}(a,b)\log(\alpha_{ab}) - \sum_{a\in\Sigma}\beta_a\left(\sum_{b\in\Sigma}\alpha_{ab} - 1\right)$$

comme la solution du problème $(\bar{\boldsymbol{\alpha}},\bar{\boldsymbol{\beta}})$ correspond à un point selle du lagrangien :

$$\frac{\partial \mathrm{L}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_{ab}} = \frac{N_{\mathbf{S}}(ab)}{\bar{\alpha}_{ab}} - \bar{\beta}_a = 0 \qquad \forall\, a,b \in \Sigma^2$$
$$\frac{\partial \mathrm{L}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \beta_a} = \sum_{b\in\Sigma}\bar{\alpha}_{ab} - 1 = 0 \qquad \forall a,b \in \Sigma^2$$

Ces relations nous donnent donc l'estimateur du maximum de vraisemblance sur une séquence **S** d'une chaîne de Markov d'ordre 1 :

$$\forall\,(a,b)\in\Sigma^2\ \hat{\Pi}(a,b) = \frac{N_{\mathbf{S}}(ab)}{N_{\mathbf{S}}(a\bullet)}$$

La forme de l'estimateur du maximum de vraisemblance nous amène à déduire que, dans la pratique, une chaîne de Markov est toujours irréductible. En effet, soit $\hat{\Pi}$ la matrice de transition estimée sur une séquence **S**. Supposons que a soit un état absorbant de la CM associée à $\hat{\Pi}$ ce qui implique : $\forall\, b\in\Sigma,\ \hat{\Pi}(a,b) = 0 \Leftrightarrow \forall\, b\in\Sigma,\ N_S(a,b) = 0$. Alors, toutes les occurrences de a devraient nécessairement apparaître en fin de séquence, ce qui entrerait en conflit avec l'hypothèse de stochasticité des séquences.

De même, les CM estimées seront toujours apériodiques. En effet, pour qu'une chaîne estimée soit périodique, il faudrait que celle ci soit composée uniquement de répétitions. Par exemple, la séquence `([AT][CG])+` mènerait à l'estimation d'une chaîne périodique de période 2.

Notons aussi que la démonstration s'appuie sur le fait que la distribution initiale μ_0 n'est pas à estimer. Si on suppose que μ_0 est la distribution stationnaire, il devient nécessaire de tenir compte du fait que μ_0 est fonction de Π. Ceci peut rendre la maximisation exacte de la vraisemblance problématique. Néanmoins, dès que l est suffisamment grand, $\log\mu_0$ devient négligeable devant les $N_{\mathbf{S}}(ab)$ et on procède alors par un simple rapport de comptages.

Ce résultat s'étend aisément au cas des modèles de Markov d'ordre k. L'estimateur du maximum de vraisemblance pour un modèle d'ordre k sur une séquence **S** s'écrit donc :

$$\forall\,(a_1,\ldots,a_m)\in\Sigma^k, \forall b\in\Sigma,\quad \hat{\Pi}(a_1\ldots a_m,b) = \frac{N_{\mathbf{S}}(a_1\ldots a_m b)}{N_{\mathbf{S}}(a_1\ldots a_m\bullet)}$$

2.1.2.2 Information de Fisher et comportement asymptotique des estimateurs

Intéressons nous maintenant au voisinage asymptotique de l'estimateur du maximum de vraisemblance. Remarquons dans un premier temps que $\mathcal{L}(\mathbf{S};\Pi)$ est continue

et dérivable en Π, on a aussi comme relation :

$$\left.\frac{\partial \mathcal{L}(\mathbf{S};\Pi)}{\partial \Pi(a,b)}\right|_{\Pi=\Pi^*} = 0 \quad \forall (a,b) \in \Sigma^2$$

Ainsi, intuitivement on peut penser que la courbure de la vraisemblance autour de $\boldsymbol{\theta}^*$ pourra donner une idée de la dispersion de $\hat{\boldsymbol{\theta}}$. Introduisons les matrices d'information de Fisher attendues et observées, qui résument cette information et permettent de quantifier la répartition asymptotique des $\hat{\mathbf{\Pi}}$ autour du vrai paramètre $\mathbf{\Pi}^*$.

Définition 2.11. Soit X une v.a. de densité $f(\cdot,\boldsymbol{\theta})$ et de fonction de vraisemblance $L(x;\boldsymbol{\theta})$ avec $\boldsymbol{\theta} \in \mathbb{R}^d$. On définit la matrice d'information attendue de Fisher $\mathcal{J}(\boldsymbol{\theta})$ comme la matrice à d lignes et d colonnes telle que :

$$\begin{aligned}[\mathcal{J}(\boldsymbol{\theta})]_{i,j} &= \mathbb{E}_\theta\left(-\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log L(X;\boldsymbol{\theta})\right) \\ &= \mathbb{E}_\theta\left(\frac{\partial}{\partial\theta_i}\log L(X;\boldsymbol{\theta})\cdot\frac{\partial}{\partial\theta_j}\log L(X;\boldsymbol{\theta})\right) \qquad \forall i,j=1,\dots,d\end{aligned}$$

Dans le cas de n réalisations indépendantes $X_1,\dots,X_n$ on note :

$$[\mathcal{J}_n(\boldsymbol{\theta})]_{i,j} = \mathbb{E}_\theta\left(-\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log L(X_1,\dots,X_n;\boldsymbol{\theta})\right)$$

Définition 2.12. Soient x une réalisation de la v.a. X de densité $f(\cdot,\boldsymbol{\theta})$ et de fonction de vraisemblance $L(\cdot;\boldsymbol{\theta})$ avec $\boldsymbol{\theta} \in \mathbb{R}^d$. On définit la matrice d'information observée de Fisher $\mathcal{I}(x,\boldsymbol{\theta})$ par :

$$[\mathcal{I}(x,\boldsymbol{\theta})]_{i,j} = -\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log L(x;\boldsymbol{\theta}) \qquad \forall i,j=1,\dots,d$$

Ces quantités permettent de décrire le comportement asymptotique des estimateurs par maximum de vraisemblance, en indiquant leur dispersion moyenne autour du vrai paramètre. En particulier, l'inégalité de Cramér-Rao fait intervenir $\mathcal{J}(\boldsymbol{\theta})$.

Dans le cas de variables indépendantes et identiquement distribuées, on a le résultat suivant de normalité asymtotique de $\hat{\theta}$:

Proposition 2.13. *Soit $y_1,\dots,y_n$ n réalisations i.i.d d'une v.a. Y de densité $f(\cdot,\boldsymbol{\theta}^*)$. Notons $\hat{\boldsymbol{\theta}}_n$ l'estimateur du maximum de vraisemblance de θ^* associé à un échantillon de taille n. On dispose des deux résultats asymptotiques suivant :*

– *Supposons f dérivable 3 fois et dont la dérivée tierce est d'intégrale bornée au voisinage de θ^*, et $\hat{\boldsymbol{\theta}}$ consistant. Alors, $\hat{\boldsymbol{\theta}}_n$ converge en loi vers une gaussienne de moyenne θ^* et de matrice de variance covariance $\frac{1}{\sqrt{n}}\mathcal{J}^{-1}(\theta^*)$. En d'autres termes :*

$$\sqrt{n}(\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_n) \xrightarrow[n\to\infty]{L} \mathcal{N}(0,\mathcal{J}^{-1}(\theta^*)) \tag{2.4}$$

– *La loi a posteriori sur $\boldsymbol{\theta}$ converge vers une gaussienne centrée en $\hat{\theta}$ et de variance $\frac{1}{\sqrt{n}}\mathcal{I}^{-1}(y_1,\ldots,y_n,\hat{\boldsymbol{\theta}})$. Notant, $\boldsymbol{\theta}_{|y_1,\ldots,y_n}$ la loi a posteriori de $\boldsymbol{\theta}$ connaissant $y_1,\ldots,y_n$:*

$$\boldsymbol{\theta}_{|y_1,\ldots,y_n} \xrightarrow[n\to\infty]{L} \mathcal{N}(0,\mathcal{I}^{-1}(\hat{\theta})) \tag{2.5}$$

Ce résultat s'étend au cas multidimensionnel en adaptant les conditions de régularité sur f. Les deux démonstrations font intervenir la formule de Taylor, au voisinage de $\boldsymbol{\theta}^*$ (eq (2.4)) ou de $\hat{\boldsymbol{\theta}}$ (eq (2.5)) suivant l'approximation considérée. Notons que dans la pratique, on ne dispose pas de $\boldsymbol{\theta}^*$ et on substitue donc $\mathcal{J}(\hat{\boldsymbol{\theta}})$ à $\mathcal{J}(\boldsymbol{\theta}^*)$. On pourra se reporter par exemple à [Azz96] pour plus de précisions.

Le principe de la démonstration précédente s'étend aux Chaînes de Markov après avoir démontré que l'estimateur par maximum de vraisemblance est consistant et que la dérivée tierce de la log-vraisemblance est bien bornée - [Bil61].

Une autre quantité d'intérêt est le vecteur des scores de Fisher, qui permet de statuer sur l'ajustement de chacun des paramètres pour un modèle donné. Ainsi, pour des paramètres $\mathbf{\Pi}_0$ et une sequence observée $\mathbf{S}_0$, on le définit par le vecteur :

$$\overrightarrow{\mathcal{V}_{\mathcal{F}}}(\mathbf{S}_0,\mathbf{\Pi}_0) = \left(\frac{\partial\mathcal{L}(\mathbf{S}_0;\mathbf{\Pi}_0)}{\partial\mathbf{\Pi}_0(a,b)}\right)_{(a,b)\in\Sigma^2}$$

Cette quantité peut être utilisée dans une perspective de classification, comme on le verra dans le chapitre suivant (section 3.3.4.2). Dans le cas d'une chaîne de Markov Π et pour une séquence observée $\mathbf{S}$, le score de Fisher se calcule comme :

$$\overrightarrow{\mathcal{V}_{\mathcal{F}}}(\mathbf{S},\mathbf{\Pi}) = \left(\frac{N_{\mathbf{S}}(a,b)}{\mathbf{\Pi}(a,b)}\right)_{(a,b)\in\Sigma^2}$$

avec la contrainte $\forall\, a\in\Sigma, \sum_{b\in\Sigma}\Pi(a,b)=1$.

2.1.3 Applications en bioinformatique

Le cadre théorique des chaînes de Markov a connu plusieurs champs d'application, des modélisations de files d'attentes au développement de stratégies optimales pour le Monopoly, [AB72].

Dans le contexte de l'analyse des séquences biologiques, cette approche a été initiée par [Cow91] visant à analyser les mots statistiquement surreprésentés dans les séquences biologiques. La justification de cette méthodologie découle de la constatation que, sans contrainte de pression de sélection, les séquences d'ADN évoluent de manière aléatoire par mutation,insertion ou délétion. On fait ainsi l'hypothèse qu'une séquence ayant évoluée sans contrainte de sélection naturelle est une chaîne de Markov. L'intérêt d'une telle hypothèse est qu'elle permet, dans le cas d'une CM d'ordre *k*, de prendre

en compte les fréquences attendues des mots de $(k+1)$ lettres sous le modèle nul. On préfère aussi généralement l'hypothèse markovienne à la multinômiale en raison de l'influence du contexte pour les mutations apparaissant à une position donnée.

Alors, les motifs dont le nombre d'occurrences dans la séquence est supérieur à l'attendu sous le modèle supposé correspondent vraisemblablement à un motif conservé en raison de son importance fonctionnelle. Initialement, cette approche a été proposée sur des génomes bactériens complets, pour la détection de sites fonctionnels. Cependant, sous l'hypothèse simplement markovienne, les résultats obtenus ne sont pas encore pleinement satisfaisants. Ceci pourrait être expliqué par le fait que le long d'un génome, l'hypothèse d'homogénéité n'est pas assez respectée.

En revanche, cette méthodologie a déjà prouvé son efficacité en étant appliquée à l'analyse des régions de régulation, ou, plus précisément, la détection des sites de fixation des facteurs de transcription au sein de séquences. Dans ce cas, le jeu de séquences étudiées est généralement restreint aux séquences en amont de gènes corrégulés. Cette sélection fonctionnelle des séquences favorise l'enrichissement en motifs d'intérêts et explique ainsi en partie la qualité de certains résultats.

Posons donc le cadre formel pour ce problème avant de voir comment on peut répondre à cette question lors de l'étude des régions de régulation. On considère une séquence d'ADN $\mathbf{S} = (s_1 s_2 \dots s_\ell)$ supposée générée par une CM d'ordre 1 de matrice de transition Π et de loi stationnaire μ.

Pour un mot $w = w_1 w_2 \dots w_{|w|}$, notons $n_{\mathbf{S}}(w)$ le nombre d'occurrences observées et $N_\ell(w)$ la variable aléatoire du nombre d'occurrences dans une séquence markovienne de longueur ℓ. On veut alors calculer :

$$\mathbf{P}(N_\ell(w) \geqslant n_{\mathbf{S}}(w)) \tag{2.6}$$

i.e. obtenir une p-valeur sur la surreprésentativité du mot w par rapport au nombre attendu sous un modèle de Markov.

Cependant, on est souvent plus intéressé par la significativité de motifs, qui correspondent par exemple à une expression régulière. Ici, on note un motif $\mathcal{W}$ comme le groupe de mots $\left\{w^1, w^2, \dots, w^{|\mathcal{W}|}\right\}$ et on étudiera la variable aléatoire $N_\ell(\mathcal{W}) = N_\ell(w^1) + N_\ell(w^2) + \dots + N_\ell(w^{|\mathcal{W}|})$.

Le calcul de ces probabilités a été abordé par diverses méthodes : (1) calcul des deux premiers moments après approximation gaussienne, (2) approximations par la loi de Poisson, (3) calcul exact par l'utilisation de formules de récurrence, (4) méthodes par grande déviation. Pour des revues sur le sujet on pourra se reporter à l'article de [RSW00] ou l'étude de [Nue] qui analyse aussi les problèmes d'instabilité numérique pouvant survenir lors des calculs.

Ici, nous présentons une manière de calculer ces probabilités à l'aide de relations de récurrence entre les probabilités d'apparition des motifs à des positions données. Cette méthode a d'abord été été proposée pour le problème du temps d'attente pour un mot

dans une séquence aléatoire (à ce sujet, se référer au [Fel68] ou l'article de [BT82]). Les résultats ont été étendus au cas de séquences markoviennes et pour des motifs par [RD99, RD00].

2.1.3.1 Probabilités d'occurrence de motifs par des méthodes exactes.

Cette méthode fait intervenir les lois d'occurrence d'un motif en une position donnée de la séquence. Introduisons les quantités nécessaires pour le calcul.

Définition 2.14. Soit $\mathcal{W} = \left(w^i\right)_{i=1,\dots,|\mathcal{W}|}$ un groupe de mot, $w = (w_1 \dots w_{|w|})$ un mot et $\mathbf{S} = s_1 \dots s_l$ une séquence markovienne. On définit les notations suivantes :

- $\{w$ apparaît en $x\}$ si le mot w **finit** en x dans $\mathbf{S}$, *i.e.* si $w_1 \cdots w_{|w|} = s_{x-|w|+1} \cdots s_x$
- $X_{w^i}(\mathcal{W})$: le temps d'attente avant la première occurrence du mot w^i dans $\mathbf{S}$ sans qu'aucun autre mot de $\mathcal{W}$ ne soit apparu avant. Quand il n'y a pas d'ambigüité sur $\mathcal{W}$, on le note simplement X_{w^i}. On note sa loi $\mathbf{p}_{w^i}$.
- $X_{w^i,w^j}(\mathcal{W})$: la distance entre w^i et w^j sans qu'aucun autre mot de $\mathcal{W}$ ne soit apparu avant, de loi $\mathbf{p}_{w^i,w^j}$. Ici, la distance entre deux mots est définie par la distance entre la fin de ces mots. Si $|w^j| < |w^i|$, on pose $\{X_{w^i,w^j} = x\} = \emptyset$ pour $x < |w^j| - |w^i|$, évitant ainsi tout problème avec des événements dont on ne saurait calculer la probabilité.
- Y_{w^i,w^j} : la distance entre w^i et w^j, de loi $\mathbf{q}_{w^i,w^j}$.
- Enfin, $X_{w^i}[n](\mathcal{W})$ correspond à l'événement où la $n^{\text{ème}}$ occurrence d'un mot de $\mathcal{W}$ est w^i. On note sa loi $\mathbf{p}_{w^i}[n]$. Bien sûr, $X_{w^i}[1] = X_{w^i}$ et on peut écrire :

$$X_{w^i}[n] = \bigcup_{i=1}^{|\mathcal{W}|} (X_{w^i}[n-1] + X_{w^j,w^i}) \tag{2.7}$$

La figure 2.4 illustre les valeurs prises par les différentes variables aléatoires sur une séquence d'ADN observée.

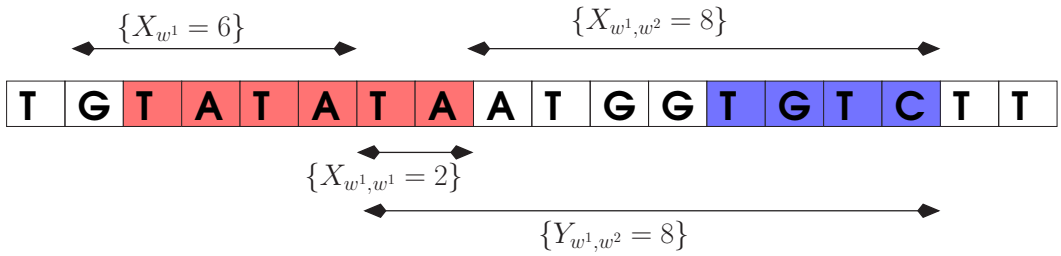


FIG. 2.4 – illustration des quantités X_w, X_{w^i,w^j} et Y_{w^i,w^j} sur une séquence d'ADN. $\mathcal{W}$ ={TATA,TGTC }.

Remarquons que le problème du calcul de la p-valeur (2.6) se ramène à la connaissance de la loi des $X^n_{w_i}$. En effet, on a :

$$\left\{N_\ell(\mathcal{W}) \geqslant n_{\mathbf{S}}(\mathcal{W})\right\} = \bigcup_{x=1}^{\ell} \bigcup_{i=1}^{|\mathcal{W}|} \{X_{w_i}[n_{\mathbf{S}}(\mathcal{W})] = x\} \tag{2.8}$$

$$\mathbf{P}(N_\ell(w) \geqslant n_{\mathbf{S}}(w)) = \sum_{x=1}^{l} \sum_{i=1}^{|\mathcal{W}|} \mathbf{p}_{w_i}[n_{\mathbf{S}}(\mathcal{W})](x) \tag{2.9}$$

On s'attachera donc dans cette section à donner des formules de récurrence permettant de calculer les valeurs $\mathbf{p}_{w_i}[n](x)$

Revenons à la figure 2.4, on voit que le mot $TATA$, de par ses propriétés d'autorecouvrement, ne permet pas les même événements d'apparition que le mot $TGTC$. Ces caractéristiques de recouvrement d'un mot ou de plusieurs mots composant un motif, influent donc sur la loi d'apparition et devront être pris en compte dans le calcul des $\mathbf{p}_{w_i}[n]$.

Définition 2.15. On définit pour un mot w l'indicateur de recouvrement ε_w comme :

$$\varepsilon_w(u) = \mathbb{I}_{\{(w_{|w|-u+1} \dots w_{|w|}) = (w_1 \dots w_u)\}} \quad \text{pour} \quad u = 1, \dots, |w| - 1$$

Pour deux mots w^i et w^j non inclus l'un dans l'autre, cette définition devient :

$$\varepsilon_{w^i,w^j}(u) = \mathbb{I}_{\{(w^i_{|w_i|-u+1} \dots w^i_{|w_i|}) = (w^j_1 \dots w^j_u)\}} \qquad \text{pour} \quad u = 1, \dots, |w_i| \wedge |w_j|$$

$$\text{en notant} \quad w^i = (w^i_1 \dots w^i_{|w_i|})$$

Comme la séquence est supposée être à l'état stationnaire, la probabilité d'apparition d'un mot w à une position donnée, notée $\mu(w)$, s'écrit :

$$\mu(w) = \mu(w_1) \prod_{x=2}^{|w|} \Pi(w_{x-1}, w_x)$$

En outre, on introduit $\tau_w(x)$, la probabilité d'observer la fin du mot w sachant que le préfixe de longueur $x-1$ est déjà apparu. En d'autre termes : $\tau_w(x) = \prod_{z=x}^{|w|} \Pi(w_{z-1}, w_z)$

Théorème 2.16 (Loi de première apparition d'un mot ou d'un motif). *Soit $w = (w_1 \dots w_k)$ un mot de longueur $|w|$, alors la loi de X_w est donnée par la relation de récurrence :*
Pour $x \geqslant |w|$

$$\begin{aligned}\mathbf{P}\{X_w = x\} = \mathbf{p}_w(x) = \mu(w) &- \sum_{z=|w|}^{x-|w|} \mathbf{p}_w(z)\Pi^{(x-z-|w|+1)}(w_{|w|}, w_1)\tau_w(2, |w|) \\ &- \sum_{z=x-|w|+1}^{x-1} \mathbf{p}_w(z)\varepsilon_w(|w| - x + z)\tau(|w| - x + z + 1, |w|)\end{aligned} \tag{2.10}$$

Soit $\mathcal{W} = (w^1, w^2, \ldots, w^{|\mathcal{W}|})$ un groupe de mots. La loi de $X_{w_i}(\mathcal{W})$ est donnée par la relation de récurrence :

$$\mathbf{P}\{X_{w^i}(\mathcal{W}) = x\} = \mathbf{p}_{w^i}(x) = \mu(w^i) - \sum_{z=|w^i|}^{x-|w^i|} \sum_{j=1}^{|\mathcal{W}|} \mathbf{p}_{w^j}(z)\Pi^{(x-z-|w^i|+1)}(w^j_{|w^j|}, w^i_1)\tau_{w^i}\left(2, |w^i|\right)$$
$$- \sum_{z=x-|w^i|+1}^{x-1} \sum_{j=1}^{|\mathcal{W}|} \mathbf{p}_{w^j}(z)\varepsilon_{w^j,w^i}(|w^i| - x + z)\tau(|w^i| - x + z + 1, |w^i|) \tag{2.11}$$

Démonstration. Cette démonstration est similaire à celle donnée dans les articles de [BT82] et [RD99]. On procède en décomposant l'événement $\{w$ apparaît en $x\}$ qui a pour probabilité $\mu(w)$. Si w apparaît en x, c'est que :

1. Soit w apparaît en x pour la première fois (soit $\{X_w = x\}$) avec la probabilité $\mathbf{p}_w(x)$.
2. Soit il existe $z < x$ tel que $X_w = z$. Celà s'écrit comme la réunion disjointe pour $z < x$ des événements $\{X_w = z\,;\, W$ apparaît en $x\}$

Dans le deuxième cas, suivant la valeur de z, la probabilité peut s'écrire :

- $z \leqslant x - |w|$: Il n'y a pas de chevauchement et le processus doit passer de $w_{|w|}$ à w_1 en $(x - z - |w| + 1)$ étapes, puis générer w à nouveau. La probabilité s'écrit donc : $\mathbf{p}(z)\Pi^{x-z-|w|+1}(w_{|w|}, w_1)\tau(2, |w|)$.
- $x - |w| \leqslant z \leqslant x - 1$: Si le mot peut se chevaucher (soit $\varepsilon_w(|w| - x + z) = 1$), il suffit alors de générer les $x - z$ dernières lettres de w. La probabilité s'écrit donc : $\mathbf{p}(z)\varepsilon_w(|w| - x + z)\tau(|w| - x + z + 1, |w|)$.

En sommant ces termes sur z, on arrive à l'égalité annoncée.
L'extension au cas d'un groupe de mots s'obtient directement en remarquant que les événements $\{X_{w^j} = z\}$ sont disjoints pour $j = 1, \ldots, \mathcal{W}$ □

Pour le calcul des probabilités de $n^{\text{ième}}$ apparition, on peut remarquer que la formule (2.7) vaut pour des événements disjoints et donc écrire une relation de récurrence sur le nombre d'occurrences en fonction de la loi des p_{w^i,w^j} :

$$\mathbf{p}_{w_i}[n](x) = \sum_{z=1}^{x-1} \sum_{j=1}^{|w|} \mathbf{p}_{w_j}[n-1](z) \cdot \mathbf{p}_{w_j,w_i}(x - z + 1)$$

la loi des p_{w^i,w^j} s'obtenant suivant le même principe que pour le théorème 2.16 :

$$\mathbf{p}_{w_i,w_j}(x) = \mathbf{q}_{w_i,w_j}(x) - \sum_{k=1}^{\mathcal{W}} \sum_{z=1}^{x-1} \mathbf{q}_{w_i,w_k}(z)\mathbf{p}_{w_k,w_j}(x - z) \tag{2.12}$$

et

$$\begin{aligned}\mathbf{q}_{w_i,w_j}(x) &= \varepsilon_{w_i,w_j}(|w_j|-x)\,\tau_{w_j}(|w_j|-x+1,|w_j|)\,\mathbb{I}_{\{x<|w_j|\}} \\ &\quad + \Pi^{(x-|w_j|+1)}(w^i_{|w_i|},w^j_1)\,\tau_{w_j}(2,|w_j|)\,\mathbb{I}_{\{x\geqslant|w_j|\}} \end{aligned} \tag{2.13}$$

Pour déterminer la loi de $\mathbf{p}_{w^i}$ sur une séquence de longueur ℓ la complexité est donc en $o\left(n\cdot|\mathcal{W}|\cdot\ell^2\right)$. En combinant avec (2.12), on voit que la complexité pour calculer la loi de $\mathbf{p}_{w^i}[n]$ devient en $o(|\mathcal{W}|^2\cdot\ell^2)$. Ce calcul devient impraticable dès que le nombre d'occurrences ou que la longueur de la séquence deviennent relativement grands.

Cependant, on peut gagner un premier ordre de grandeur sur la longueur en utilisant la vitesse de convergence exponentielle des Π^x vers la loi stationnaire. En effet (théorème 2.10), on peut fixer une valeur $\alpha\in\mathbb{N}$, dépendant d'une précision voulue ε et de la seconde valeur propre de Π, telle que $\|\Pi^\alpha-\mu\|<\varepsilon$. Les relation de récurrence (2.10,2.11) ne portent plus que sur au plus α termes. Pour plus de détails concernant les considérations numériques que nécessite ce type d'implémentation, on pourra se référer à [Nue].

2.1.3.2 Application à la détection de sites de fixation des facteurs de transcription

Revenons sur le problème d'apprentissage posé en page 37. Les technologies actuelles basées sur les expériences de puces à ADN permettent d'extraire des familles de gènes exprimées dans les mêmes conditions expérimentales. Une analyse plus fine permet en particulier de restreindre ces familles à des groupes de gènes dont l'expression est régulée par un même facteur de transcription. Connaissant cette famille de gènes corrégulés, on souhaite alors identifier les motifs correspondant aux sites de fixations du facteur de transcription.

Plus formellement, on a un jeu de séquences $\mathbf{S}_1,\mathbf{S}_2,\ldots,\mathbf{S}_n$ défini par les contraintes suivantes :

- $\mathbf{S}_1,\mathbf{S}_2,\ldots,\mathbf{S}_n$ sont les régions en amont d'un groupe de gènes $g_1,\ldots,g_n$, et incluant leurs régions régulatrices.
- l'expression des gènes $g_1,\ldots,g_n$ est régulée par au moins un facteur de transcription en commun.

Le problème posé est alors d'identifier le ou les motifs communs au groupe de séquences.

Cette problématique est très riche en bioinformatique, et a suscité la mise au point de nombreuses méthodes, tant algorithmiques que statistiques. Une approche populaire consiste à modéliser le site de fixation par une matrice poids/position (Position Specific Scoring Matrix ou PSSM). Ce modèle suppose alors que la loi d'apparition des lettres à chaque position du site de fixation est une loi multinomiale. L'apprentissage consiste ensuite à estimer l'ensemble des paramètres "matrice" et "positions des sites" qui maximisent la vraisemblance sur le jeu de séquence. Nous ne traiterons pas ce cas mais il a

donné lieu à de nombreux outils fréquemment utilisés (comme par exemple MEME ou le Gibbs Sampler).

Une approche complémentaire a été proposée en travaillant sur les mots surreprésentés au sein d'une séquence. La présentation qui suit est largement inspirée par une étude de Van Helden et al. [vHACV98] portant sur les régions régulatrices de dix familles de gènes corrégulés chez la levure.

Nous supposerons par la suite que pour tout motif $\mathcal{W}$, $n \in \mathbb{N}$ et une matrice de transition Π connue, la p-valeur $\mathbf{P}(N_\ell(\mathcal{W}) \geq n)$ est calculable en un temps raisonnable. En outre, afin de mettre en avant les différences d'ordre de grandeur pour les p-valeurs, on travaillera par la suite en échelle logarithmique.

La connaissance du mécanisme biologique d'initiation de la transcription permet de rajouter un certain nombre d'hypothèses pour l'étude sur la levure :

- 99% des sites déterminés expérimentalement sont positionnés à moins de 800 bases en amont du codon start. Logiquement, l'étude devrait porter sur les régions en amont du site d'initiation de la transcription, mais leur détermination expérimentale ou automatique est insuffisante ou peu fiable. $\mathbf{S}_1, \dots \mathbf{S}_n$ correspondent donc aux régions constituées des 800 bases en amont du codon start de $g_1, \dots, g_n$.
- Le site de fixation doit généralement apparaître en plusieurs exemplaires dans la séquence pour être fonctionnel. On étudiera donc le nombre d'occurences totales des mots dans $\mathbf{S}_1, \dots, \mathbf{S}_n$ dont on notera simplement $p_\Pi(w)$ la $p$valeur sous la CM de matrice de transition Π
- Les sites de fixations peuvent être actifs sur les deux brins de l'ADN. Afin de prendre en compte cette information, si w n'est pas palindromique, on étudiera les motifs de type $\{w, \overline{w}\}$ où $\overline{w}$ représente l'inverse complémentaire de w.

La matrice de transition $\hat{\Pi}$ est estimée sur l'ensemble des régions intergéniques de la levure, en prenant un modèle d'ordre $k-1$ lors de l'étude des k-mers (les mots de longueur k). Ensuite, on procède au calcul de $p_{\hat{\Pi}}\{w, \overline{w}\}$ pour tous les mots w de longueur k. Remarquons qu'on teste alors $D_k = 4^k - \frac{1}{2}4^k + 4^{w/2}$ motifs ($4^{w/2}$ correspondant au nombre de palindromes possibles). On peut vouloir appliquer une correction sur ce multitest pour corriger les significativités. En appliquant une correction de Bonferroni, l'indice de significativité regardé est alors :

$$sig_{(w,\overline{w})} = -\log_{10}\left(D_k \cdot p_\Pi(w, \overline{w})\right)$$

La figure 2.5 montre les résultats obtenus sur 5 des 10 familles, en procédant à l'étude des hexanucléotides, et en regroupant les motifs significatifs en clusters. On peut voir que, à l'exception de la famille GAL, où aucun motif n'est extrait, le cluster possédant les motifs les plus significatifs correspond à celui annoté dans la littérature.

Dans le cas de la famille GAL, le facteur de transcription correspondant possède comme motif de liaison à l'ADN `CGGN{5}WN{5}CCG`, où les deux régions déterminées sont donc séparés de 11 bases. Ce type de motifs ne peut être détecté par une approche

utilisant des hexanucléotides. On peut par contre étendre les motifs étudiés à ceux composés de deux motifs séparées par une distance variable. Ces contraintes rendent le calcul des probabilités d'occurrence plus compliqué, et peuvent nécessiter l'utilisation de simulations (voir par exemple [VHRCV00] ou [MS00, VMS99] pour les applications à la détection de promoteurs chez les eucaryotes).

Insistons aussi sur l'importance du modèle utilisé lors de l'estimation des paramètres, et plus particulièrement sur l'intérêt d'utiliser des modèles de Markov d'un ordre suffisant. En effet, la table 2.1 montre les différences de classement qui surviennent lors de l'utilisation de différents modèles, de la loi multinomiale où toutes les lettres sont équiprobables, aux CM d'ordre 5. Les différences de classement sur les motifs d'intérêt justifient l'attention qui doit être portée lors du choix du modèle de background, tant pour les données servant à l'estimation que dans le choix de l'ordre de la CM.

Family	Pattern	r1	r2	r3
MET	**CACGTG**	**1**	**23**	**5**
	TCACGT	2	21	6
	GTCACG	6	259	24
	TGTGGC	8	131	12
	CTGTGG	**3**	**99**	**8**
	ACTGTG	7	86	17
	AACTGT	4	32	18
	ATATAT	5	3	2
	TATATA	10	1	3
	GCTTCC	9	75	7

TAB. 2.1 – tiré de [vHdOPO00], comparaison du classement des 6-mers sur les régions de régulation de la famille MET. Le rang est comparé pour les motifs surreprésentés pour le modèle de la figure 2.5 (r1), dans un modèle multinomiale où toutes les lettres sont équiprobables (r2) et pour un modèle M0 estimé sur les séquences intergéniques (r3). Les motifs connus expérimentalement sont indiqués en gras.

Si l'analyse des mots surreprésentés possède de bonnes performances en étant appliquée à des organismes simples, l'extension aux eucaryotes supérieurs, par exemple les mammifères, pose des problèmes supplémentaires. En effet, les régions de régulations sont alors de plus grande taille (au minimum 3000 paires de bases) et présentent alors une composition hétérogène le long de la séquence, remettant ainsi en cause l'utilisation d'un modèle markovien comme hypothèse nulle. Ainsi, des motifs sans lien avec le site de fixation posséderont une forte statistique en raison d'un biais de composition locale.

Family	Hexanucleotide analysis result Sequence	Ms	Occ	Exp	*sig*	Sites previously characterized Consensus	Bound factors
NIT	ATAAGA	6	20	6.0	**2.0**		Gln3p, Nil1p,
	GATAAG	6	26	3.0	9.1	GATAAG	Gzf3p, Uga43p
	AGATAA	7	17	6.1	0.4		(Zn finger)
	CTGATA	6	11	3.1	0.1		
	CCGCGC	2	6	0.7	0.8	–	–
	CGGCAC	4	6	0.8	0.5	–	–
	ACATCT	4	11	2.9	0.4	–	–
MET	CACGTG	9	26	2.0	**7.0**		Cbf1p-Met4p-
	TCACGT	9	19	2.9	**6.1**	TCACGTG	Met28p complex
	GTCACG	6	8	1.4	0.7		(bHLH-bLZ-bLZ)
	TGTGGC	7	10	2.4	0.5		
	CTGTGG	8	11	2.1	**1.6**	AAAACTGTGG	Met31p, Met32p
	ACTGTG	9	12	3.2	0.6		(Zn finger)
	AACTGT	10	17	5.5	0.9		
	ATATAT	19	82	42.3	0.8	–	–
	TATATA	11	80	43.9	0.2		
	GCTTCC	7	12	3.5	0.2	–	–
PHO	CGCACG	5	6	0.5	**1.5**		
	GCACGT	5	10	0.8	**4.4**		Pho4p
	CACGTG	5	12	0.9	**1.8**	GCACGTGGG	(bHLH)
	ACGTGG	5	8	0.7	**2.8**	(high affinity)	
	CGTGGG	3	5	0.5	0.5		
	CACGTT	5	7	1.2	0.3	GCACGTTTT	Pho4p
	ACGTTT	5	11	2.6	0.8	(medium affinity)	(bHLH)
	CTGCAC	4	8	1.0	**1.7**	–	–
	TGCCAA	4	12	2.0	**2.6**	–	–
PDR	TCCGTG	5	8	1.1	**1.4**		
	CCGTGG	4	12	1.1	**7.4**		
	CGTGGA	5	10	1.1	**3.3**		
	GTGGAA	6	11	2.8	0.5		
	TCCGCGG	3	10	0.8	**4.5**		Pdr1p, Pdr3p
	CCGCGG	2	12	0.6	**2.6**	TCCGCGGA	(Zn_2Cys_6
	CGCGGA	3	10	0.8	**4.5**		binuclear
	CTGCGG	2	6	0.9	0.2		cluster)
	GCGCGA	5	6	0.8	0.6	–	–
	AGGCAC	3	7	1.3	0.1	–	
	GGCACC	5	6	0.9	0.2		–
GAL	–	-	-	-	-	$CGGN_3WN_5CCG$	Gal4p (Zn_2Cys_6 binuclear cluster)

FIG. 2.5 – Détection des sites de régulation par analyse statistique des hexanucléotides, tiré de [vHdOPO00]. Pour chaque famille, les motifs présentant un score $sig \geq 0$ sous un modèle markovien d'ordre 5 sont représentés (les scores supérieurs à 1 sont écrits en gras). Les motifs sont regroupés en clusters par similarité de séquence et les substitutions sont soulignées. Les deux dernières colonnes rappellent le ou les motifs déjà caractérisés. Remarquons que le motif le plus significatif correspond généralement au site annoté. Abréviations : Ms, nombre de séquences où le motif apparaît au moins une fois, occ, nombre d'occurrences du motif sur l'ensemble des séquences, exp : nombre d'occurrence attendues.

2.2 Chaînes de Markov cachées

Les modèles de Markov cachés sont définis par deux processus probabilistes générés conjointement :

- une chaîne de Markov à valeur sur un espace d'états finis $\mathcal{S}$, dont les valeurs ne sont pas observées (les états cachés)
- un jeu de fonctions aléatoires, chacune associée avec un des états de $\mathcal{S}$.

Ces modèles, présentés originellement par Baum et Petrie [Pet69, BP66] ont été utilisés dans divers domaines d'application, tel que la reconnaissance de la parole (voir à ce sujet la revue de Rabiner [Rab89]).

Au sein d'un génome, les régions codantes, intergéniques, ou de régulation, peuvent, de par la contrainte fonctionnelle imposée sur leur composition en nucléotides, être associées à des modèles de Markov différents. De même, au sein d'une protéine, plusieurs domaines peuvent se succéder, la composition en acides aminés déterminant le rôle fonctionnel de tel ou tel domaine.

L'ajustement de modèles de Markov cachés permet donc d'intégrer ce type d'information en permettant la succession de différentes CM au sein d'une même séquence.

2.2.1 Définition

On considère donc deux processus, la suite des variable aléatoires $Y_1, \ldots, Y_t$ correspondant à la séquence (à valeur dans Σ), et la suite des états cachés $S_1, \ldots, S_t$ à valeur dans un espace fini $\mathcal{S}$.

Définition 2.17 (Modèle M1-M1). Soient Σ et $\mathcal{S}$ deux ensembles de cardinal fini. Soient $\{Y_t\}_{t\geq 1}$ et $\{S_t\}_{t\geq 1}$ deux processus à valeur respectivement dans Σ et $\mathcal{S}$.
On dit que le processus $\{X_t, Y_t\}_{t\geq 1}$ est une chaîne de Markov cachée (M1-M1) sur $\Sigma \times \mathcal{S}$ si et seulement si :
$\forall\, t > 1, \forall\, y_1, \ldots, y_t, y_{t+1} \in \Sigma,$
$\forall\, s_1, \ldots, s_t, s_{t+1} \in \mathcal{S},$

$$\mathbf{P}(S_{t+1} = s_{t+1} \mid S_t = s_t, \ldots, S_1 = s_1) = \mathbf{P}(S_{t+1} = s_{t+1} \mid S_t = s_t) \qquad \text{(CM sur } \mathcal{S}\text{)} \tag{2.14}$$

$$\mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_1^t = y_1^t, S_1^{t+1} = s_1^{t+1}) = \mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_t = y_t, S_{t+1} = s_{t+1}) \tag{2.15}$$
(CM sur Σ conditionellement à $\mathcal{S}$)

De plus la chaîne est supposée homogène, *i.e.* :
$\forall\, t > 1, \forall\, u, v \in \mathcal{S}, \forall\, a, b \in \Sigma$

$$\mathbf{P}(S_{t+1} = v \mid S_t = u) = \alpha(u, v) \tag{2.16}$$

$$\mathbf{P}(Y_{t+1} = b \mid Y_t = a, S_{t+1} = u) = \Pi_u(a, b) \tag{2.17}$$

On peut étendre cette définition en permettant un ordre k_u conditionnellement chaque état u de $\mathcal{S}$.

$$\mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_1^t = y_1^t, S_1^{t+1} = s_1^{t+1}) = \mathbf{P}(Y_{t+1} = y_{t+1} \mid Y_{t-r_u+1}^t = y_{t-r_u+1}^t, S_{t+1} = s_{t+1})$$

et les Π_u ($u \in \mathcal{S}$) sont donc des matrices de transition pour des CM d'ordre k_u. Si $k_u = m$ pour tout u dans $\mathcal{S}$, on parle de modèles M1-Mm.

Les lois initiales sont généralement fixées afin de correspondre aux lois stationnaires des différentes CM :

- $S_1 \sim \alpha(\cdot)$: la loi stationnaire associée à la CM de matrice de transition $\alpha(\cdot,\cdot)$.
- $Y_1^{k_u} \mid \{S_1 = u\} \sim \mu_u$: la loi stationnaire associée à la CM de matrice de transition Π_u.

La figure 2.6 montre une réalisation d'une chaîne de Markov cachée avec les matrices ayant servi à l'apparition des lettres ou des états cachés. Comme on peut le voir de manière plus imagée, la suite des états cachés, correspondant à la suite de "couleurs", conditionne le choix de la matrice de transition et donc la loi d'apparition des lettres. On parle de données *complètes* quand on dispose de la séquence des observés (à valeur sur Σ) et de celle des états cachés (sur $\mathcal{S}$). On parlera de données *incomplètes* si la suite des états cachés n'est pas connue.

L'une des problématiques associées aux CMC est alors de reconstruire la suite des états cachés associée à une séquence biologique. En effet, dans le cas d'un génôme nouvellement séquencé, ne disposant dans une première étape que de la séquence brute, on veut pouvoir procéder à son annotation fonctionnelle, en identifiant dans un premier temps les localisations des gènes, ARN ribosomaux ou éléments de régulation.

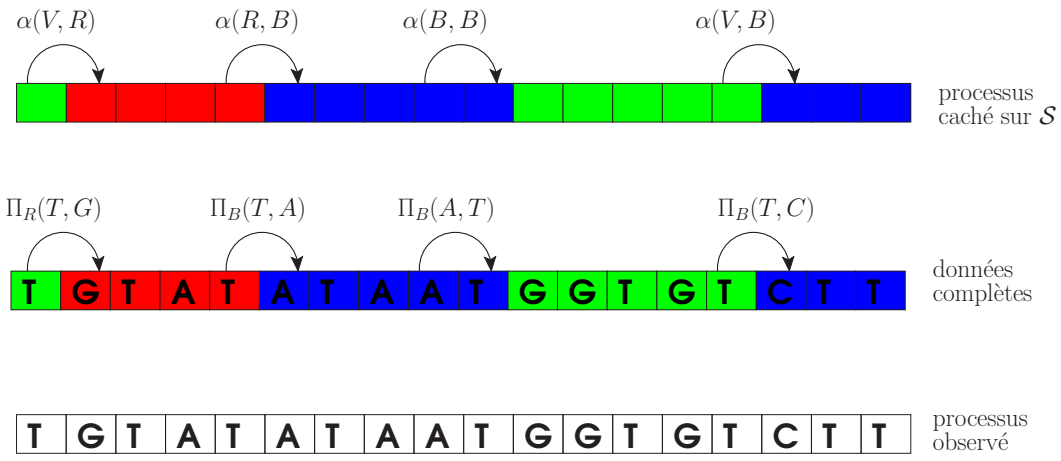


FIG. 2.6 – Réalisation d'une CMC sur Σ = {A,C,G,T} et $\mathcal{S}$ = {rouge,vert,bleu}

Calculons la vraisemblance d'une séquence **S** où les états cachés sont connus ($(y_1^t, s_1^t) \in$

$\Sigma^t \times \mathcal{S}^t$). Par conditionnements successifs, on obtient :

$$\begin{aligned}
L(\mathbf{S}, s_1^t; (\Pi_u)_{u\in\mathcal{S}}, \alpha) &= \mathbf{P}(Y_1^t = y_1^t, S_1^t = s_1^t \mid \alpha, (\Pi_u)_{u\in\mathcal{S}}) \\
&= \mathbf{P}(Y_2^t = y_2^t, S_2^t = s_2^t \mid Y_1 = y_1, S_1 = s_1)\mathbf{P}(Y_1 = y_1 | S_1 = s_1)\mathbf{P}(S_1 = s_1) \\
&= \alpha(s_1)\mu_{s_1}(y_1)\mathbf{P}(Y_3^t = y_3^t, S_3^t = s_3^t \mid Y_2 = y_2, S_2 = s_2)\cdot \\
&\qquad\qquad \mathbf{P}(Y_2 = y_2, S_2 = s_2 \mid Y_1 = y_1, S_1 = s_1) \\
&\vdots \\
&= \alpha(s_1)\mu_{s_1}(y_1)\prod_{i=2}^{t} \alpha(s_{i-1}, s_i)\Pi_{s_i}(y_{i-1}, y_i)
\end{aligned}$$

et pour la log vraisemblance :

$$\begin{aligned}
\mathcal{L}(\mathbf{S}, s_1^t; (\Pi_u)_{u\in\mathcal{S}}, \alpha) &= \log\alpha(s_1) + \log\mu_{s_1}(y_1) + \sum_{i=2}^{t}\sum_{u,v\in\mathcal{S}} \mathbb{I}_{\{s_{i-1}=u, s_i=v\}} \log\alpha(u,v) \\
&\quad + \sum_{i=2}^{t}\sum_{u\in\mathcal{S}}\sum_{a,b\in\Sigma} \mathbb{I}_{\{s_i=u\}}.\,\mathbb{I}_{\{y_{i-1}=a, y_i=b\}} \log\Pi_u(a,b)
\end{aligned}$$

Cependant, on désire en général calculer la vraisemblance avec les données incomplètes, *i.e.* avec la seule connaissance de la séquence observée ($y_1 \cdots y_t$). Dans ce cas, la vraisemblance s'écrit :

$$L(\mathbf{S}; \Pi_u, \alpha) = \sum_{s_1^t\in\mathcal{S}^t} \alpha_0(s_1)\mu_{s_1}(y_1)\left(\prod_{i=2}^{t} \alpha(s_{i-1}, s_i)\Pi_{s_i}(y_{i-1}, y_i)\right)$$

Le calcul de cette quantité possède une complexité en $o(t\cdot|\mathcal{S}|^t)$, et s'avère donc impossible dans la pratique.

2.2.2 Estimation

Dans le cas de données complètes (*i.e.* quand la suite des états cachés est connue), remarquons que la CMC peut être replongée dans une CM d'ordre m sur l'alphabet augmenté $\tilde{\Sigma} = \Sigma \times \mathcal{S}$.

Ainsi, en reprenant les résultats de la section 2.1.2, l'estimateur du maximum de vraisemblance s'écrit :

$$\begin{aligned}
\hat{\alpha}(u,v) &= \frac{N_{uv}(\bullet)}{N_{u\bullet}(\bullet)} \qquad \forall u, v \in \mathcal{S} \\
\hat{\Pi}_u(a,b) &= \frac{N_u(ab)}{N_u(a\bullet)} \qquad \forall u \in \mathcal{S}, a, b \in \Sigma
\end{aligned}$$

avec

$N_{uv}(\bullet) = \sum_{a,b\in\Sigma^2} N_{uv}(ab)$, où $N_{uv}(ab) = \sum_{i=2}^{t} \mathbb{I}\{s_{t-1} = u, s_t = v, y_{t-1} = a, y_t = b\}$, et ainsi de suite.

Nous étudierons ici les propriétés de l'estimateur du maximum de vraisemblance. Comme montré précédemment, maximiser la vraisemblance sur des données incomplètes est impossible dans la pratique. Nous présenterons donc ici une utilisation de l'algorithme EM pour l'estimation des paramètres d'une CMC.

Remarquons dans un premier temps le problème d'identifiabilité qui se présente dans le cas de données incomplètes. En effet, si σ représente une permutation de $\mathcal{S}$, on a, pour un modèle M1-M1 :

$$\begin{aligned}
\mathbf{P}(Y_1^t = y_1^t \mid \alpha, (\Pi_u)_{u\in\mathcal{S}}) &= \sum_{(s_1,\cdots,s_t)\in\mathcal{S}^t} \alpha(s_1)\,\mu_{s_1}(y_1)\left(\prod_{i=2}^{t} \alpha(s_{i-1}, s_i)\Pi_{s_i}(y_{i-1}, y_i)\right) \\
&= \sum_{(s_1,\cdots,s_t)\in\mathcal{S}^t} \alpha(\sigma(s_1))\,\mu_{\sigma(s_1)}(y_1)\left(\prod_{i=2}^{t} \alpha(\sigma(s_{i-1}), \sigma(s_i))\Pi_{\sigma(s_i)}(y_{i-1}, y_i)\right) \\
&= \mathbf{P}(Y_1^t = y_1^t \mid \sigma(\alpha), (\Pi_\sigma(u))_{u\in\mathcal{S}}) \quad (\text{avec } \sigma(\alpha(u,v)) = \alpha(\sigma(u), \sigma(v)))
\end{aligned}$$

Il est donc nécessaire de quotienter l'espace des paramètres $\boldsymbol{\theta}$ sur la relation d'équivalence notée "$\sim$" : $\boldsymbol{\theta} \sim \boldsymbol{\theta}' \Leftrightarrow \forall\, \mathbf{S}, L(\mathbf{S}, \boldsymbol{\theta}) = L(\mathbf{S}, \boldsymbol{\theta}')$. Etendant la méthode de [Pet69] des modèles M1-M0 aux modèles M1-M1, [Mur97] montre dans sa thèse que, sous certaines conditions, la classe d'équivalence pour un paramètre $\boldsymbol{\theta}$ est caractérisée par l'ensemble des paramètres obtenus par permutation sur les indices de $\mathcal{S}$.

Ces conditions sont les suivantes :

C-I : la Chaîne de Markov S_t de matrice de transition α est ergodique (*i.e.* irréductible et apériodique).

C-II : $\det\alpha \neq 0$ et $\forall\, u \in \mathcal{S}, \det(\Pi_u) \neq 0$; $\forall\, u, v \in \mathcal{S}, u \neq v, \forall\, i, j \in \Sigma$, $\Pi_u(i, j) \neq \Pi_v(i, j)$ (les matrices de transitions conditionnelles aux états sont toutes différentes terme à terme).

On supposera donc par la suite que **C**-I et **C**-II sont remplies.

2.2.2.1 Propriétés de l'estimateur du maximum de vraisemblance

Nous présentons les résultats de consistance et de normalité asymptotique de l'estimateur du maximum de vraisemblance. Ces résultats sont présentés dans le but de servir les chapitres suivants (particulièrement comme une justification des noyaux de Fisher présentés en 3.3.4.2). Le lecteur intéressé par les démonstrations pourra consulter la thèse de [Mur97].

Pour pouvoir démontrer les résultats suivants, il est nécessaire de prendre une condition plus contraignante que **C**-I.

C-III : $\boldsymbol{\theta}^*$ appartient à l'ensemble $\Theta_\delta = \{\boldsymbol{\theta} \mid \forall\, u, v \in \mathcal{S}, \forall\, a, b \in \Sigma,\ \alpha(u,v) \geq \delta, \Pi_u(a,b) \geq \delta\}$

Théorème 2.18 (convergence). *Sous la condition* **C**-*III, on a :*

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbf{P}(Y_1, \ldots, Y_n \mid \boldsymbol{\theta}) = H_{\boldsymbol{\theta}^*}(\boldsymbol{\theta}) \qquad \textit{existe p.p. sous } \boldsymbol{\theta}^*$$

En outre pour $k = 0, 1, 2, 3$ *:*

$$\lim_{n\to\infty} D^k \log \mathbf{P}(Y_0 \mid Y_{-1}, \ldots, Y_{-n+1}, \boldsymbol{\theta}) = D^k \log \mathbf{P}(Y_0 \mid Y_{-1}, \ldots, \boldsymbol{\theta})$$
$$\textit{existe uniformément en } \boldsymbol{\theta} \textit{ pour tout } Y \quad (2.18)$$

et :

$$D^k H_{\boldsymbol{\theta}^*}(\boldsymbol{\theta}) = D^k\, \mathbb{E}_{\boldsymbol{\theta}^*}(\log \mathbf{P}(Y_0 \mid Y_{-1}, \ldots, \boldsymbol{\theta})) \qquad \textit{existe}$$

Enfin, si $\hat{\boldsymbol{\theta}}_n = \arg\max_{\boldsymbol{\theta}\in\Theta_\delta} \{\log \mathbf{P}(y_1, \ldots, y_n \mid \boldsymbol{\theta})\}$ *alors* $\hat{\boldsymbol{\theta}}_n \xrightarrow{p.s} \boldsymbol{\theta}^*$ *dans la topologie quotient définie sur la relation "*$\sim$*".*

Soit :

$$\sigma_{\boldsymbol{\theta}} = \left(-\frac{\partial^2}{\partial\theta_i \partial\theta_j} H_{\theta^*}(\theta) \right)_{1\leq i,j\leq d}$$

Remarquons qu'on a $\sigma_\theta = \frac{1}{n}\mathcal{J}_n(\boldsymbol{\theta})$. On a alors comme résultats sur l'existence de solutions consistantes, et pour le comportement asymptotique :

Théorème 2.19 (normalité asymptotique). *Supposons que* $\sigma_{\boldsymbol{\theta}}$ *est définie positive en* $\boldsymbol{\theta}^*$*, alors :*

i. il existe une solution consistante des équations de vraisemblance qui maximise au moins localement la log-vraisemblance avec une probabilité qui tend vers 1 quand n tend vers l'infini.

ii. Le score de Fisher converge vers une gaussienne centrée de variance $\sigma_{\boldsymbol{\theta}^*}$ *:*

$$\frac{1}{\sqrt{n}} D \log \mathbf{P}(Y_1, \ldots, Y_n \mid \boldsymbol{\theta}^*) \xrightarrow[n\to\infty]{} \mathcal{N}(0, \sigma_{\boldsymbol{\theta}^*})$$

iii. L'écart $(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)$ *est asymptotiquement normal :*

$$\sqrt{n}\left(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*\right) \xrightarrow[n\to\infty]{} \mathcal{N}\left(0, \sigma_{\boldsymbol{\theta}^*}^{-1}\right)$$

Ceci est bien en accord avec les résultats énoncés précédemment sur les modèles indépendants et Markoviens.

2.2.2.2 Algorithme EM

L'algorithme EM est une procédure itérative et déterministe permettant de maximiser la vraisemblance d'un échantillon dans le cas de données incomplètes. En outre, sous certaines conditions, vérifiées dans le cas des CMC, l'estimateur par EM converge localement vers un paramètre satisfaisant les équations de consistance.

Comme vu précédemment, le calcul de la vraisemblance $\mathbf{P}(y_1^t \mid \boldsymbol{\theta})$ pour des données incomplètes est impraticable. Cependant, en utilisant la formule de Bayes on peut la réécrire comme :

$$\log \mathbf{P}(y_1^t \mid \boldsymbol{\theta}) = \log \mathbf{P}(y_1^t, s_1^t \mid \boldsymbol{\theta}) - \log \mathbf{P}(y_1^t \mid s_1^t, \boldsymbol{\theta})$$

Et en prenant l'espérance sous un autre paramètre $\boldsymbol{\theta}'$ conditionnellement à y_1^t :

$$\begin{aligned}\log \mathbf{P}(y_1^t \mid \boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{\theta}'}(\log \mathbf{P}(y_1^t, s_1^t \mid \boldsymbol{\theta}) \mid y_1^t) - \mathbb{E}_{\boldsymbol{\theta}'}(\log \mathbf{P}(y_1^t \mid s_1^t, \boldsymbol{\theta}) \mid y_1^t) \\ &=: Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') - H(\boldsymbol{\theta} \mid \boldsymbol{\theta}')\end{aligned}$$

Le principe de l'algorithme est alors, partant d'un paramètre $\boldsymbol{\theta}^{(0)}$, de choisir à l'étape m le paramètre $\boldsymbol{\theta}^{(m+1)}$ qui maximise $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$. En effet, par construction la suite des $Q(\boldsymbol{\theta}^{(m+1)}|\boldsymbol{\theta}^{(m)})$ est strictement croissante et, par l'inégalité de Jensen, la suite $H(\boldsymbol{\theta}^{(m+1)}|\boldsymbol{\theta}^{(m)})$ est strictement décroissante.

Cette reformulation permet surtout de travailler avec des quantités calculables. En effet, à chaque étape, on "complète" par la moyenne des données, conditionnellement à la séquence et à la valeur actuelle du paramètre :

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)}) = \sum_{s_1^t \in \mathcal{S}^t} \log \mathbf{P}(y_1^t, s_1^t \mid \boldsymbol{\theta}) \cdot \mathbf{P}(s_1^t \mid y_1^t, \boldsymbol{\theta}^{(m)})$$

et le calcul de cette quantité, une fois connue la loi des $\mathbf{P}(S_i \mid y_1^t, \boldsymbol{\theta}^{(m)})$ est linéaire sur la longueur de la séquence.

Pour résumer, la suite de paramètres $\boldsymbol{\theta}^{(m)}$ est donc générée de la manière suivante :

Etape E (pour "Expectation") : on complète les données conditionnellement à $\boldsymbol{\theta}^{(m)}$ en calculant $\forall\, u \in \mathcal{S}, i = 1, \ldots, t : \mathbf{P}(S_i = u \mid y_1^t, \boldsymbol{\theta}^{(m)})$

Etape M de Maximisation, on met a jour les paramètres en maximisant $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)})$: $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\theta \in \Theta} \left\{ \sum_{s_1^t \in \mathcal{S}^t} \log \mathbf{P}(y_1^t, s_1^t \mid \theta) \cdot \mathbf{P}(s_1^t \mid y_1^t, \theta^{(m)}) \right\}$

Etape E Le calcul des probabilités a posteriori pour les états cachés fait intervenir des formules de récurrence sur la séquence. En effet, en conditionnant successivement sur les états cachés possibles avant et après une position, on tire parti du caractère markovien des processus sur les observés et sur les états cachés. Le calcul se ramène alors à un problème de programmation dynamique résumé par les trois équations suivantes.

Equation prédictive

$$\mathbf{P}(s_i = v \,|\, y_1^{i-1}, \boldsymbol{\theta}) = \sum_{u=1}^{q} \alpha(u,v) \mathbf{P}(s_{i-1} = u \,|\, y_1^{i-1}, \boldsymbol{\theta}) \tag{2.19}$$

Equation de filtrage

$$\mathbf{P}(s_i = v \,|\, y_1^{i}, \boldsymbol{\theta}) = \frac{\Pi_v(y_{i-r_v}^{i-1}, y_i)\ \mathbf{P}(s_i = v \,|\, y_1^{i-1}, \boldsymbol{\theta})}{\sum_{u=1}^{q} \Pi_u(y_{i-r_u}^{i-1}, y_i)\ \mathbf{P}(s_i = u \,|\, y_1^{i-1}, \boldsymbol{\theta})} \tag{2.20}$$

Equations de lissage

$$\mathbf{P}(s_{i-1} = u, s_i = v \,|\, y, \boldsymbol{\theta}) = \frac{\alpha(u,v)\, \mathbf{P}(s_{i-1} = u \,|\, y_1^{i-1}, \boldsymbol{\theta})\, \mathbf{P}(s_i = v \,|\, y, \boldsymbol{\theta})}{\mathbf{P}(s_i = v \,|\, y_1^{i-1}, \boldsymbol{\theta})} \tag{2.21}$$

$$\begin{aligned} \mathbf{P}(s_{i-1} = u \,|\, y, \boldsymbol{\theta}) &= \sum_{v=1}^{q} \mathbf{P}(s_{i-1} = u, s_i = v \,|\, y, \boldsymbol{\theta}) \\ &= \mathbf{P}(s_{i-1} = u \,|\, y_1^{i-1}, \boldsymbol{\theta}) \sum_{v=1}^{q} \frac{\mathbf{P}(s_i = v \,|\, y, \boldsymbol{\theta})}{\mathbf{P}(s_t = v \,|\, y_1^{t-1}, \boldsymbol{\theta})}\ \alpha(u,v) \end{aligned} \tag{2.22}$$

L'algorithme suivant est couramment appelé "forward-backward" en raison de son caractère doublement récursif. En effet, la première récurrence, pour t croissant (étape forward) permet de calculer $\mathbf{P}(s_t = v | y_1^t, \boldsymbol{\theta})$. Ensuite, la seconde récurrence permet d'obtenir les probabilités a posteriori pour tous les états (étape backward). La complexité de cet algorithme est en $o(t \cdot |\mathcal{S}|^2)$.

Remarquons aussi que les $\mathbf{P}(s_{i-1} = u \,|\, y_1^t, \boldsymbol{\theta})$, calculés conditionnellement à $\hat{\boldsymbol{\theta}}$, fournissent ainsi une annotation sur l'état caché le plus probable en chaque lettre de la séquence. Comme on le verra dans les sections suivantes, cette information peut être utilisée pour l'annotation fonctionnelle des séquences biologiques.

Etape M Pour l'étape de maximisation, rappelons que la vraisemblance des données complètes s'écrit :

$$\begin{aligned} \log \mathbf{P}(y_1^t, s_1^t \,|\, \theta) = \log \alpha(s_1) + \log \mu_{s_1}(y_1) \\ + \sum_{i=2}^{t} \sum_{u,v \in \mathcal{S}} \mathbb{I}_{\{s_{i-1}=u, s_i=v\}} \log \alpha(u,v) \\ + \sum_{i=2}^{t} \sum_{u \in \mathcal{S}} \sum_{a,b \in \Sigma} \mathbb{I}_{\{s_i=u\}} . \mathbb{I}_{\{y_{i-1}=a, y_i=b\}} \log \Pi_u(a,b) \end{aligned}$$

Algorithm 1 Algorithme forward-backward

Requis : séquence y_1^t
paramètres $\boldsymbol{\theta} = (\alpha, \{\Pi_u, u \in \mathcal{S}\})$

(étape forward)
Pour $i = 2 \nearrow t$ **Faire**
Pour tout $u \in \mathcal{S}$ **Faire**
Calculer $\mathbf{P}(s_i = u \,|\, y_1^{i-1}, \boldsymbol{\theta})$ à l'aide de (2.19)
Fin Pour
Pour tout $u \in \mathcal{S}$ **Faire**
Calculer $\mathbf{P}(s_i = u \,|\, y_1^{i}, \boldsymbol{\theta})$ à l'aide de (2.20)
Fin Pour
Fin Pour

(étape backward)
Pour $i = t \searrow 2$ **Faire**
Pour tout $u, v \in \mathcal{S}$ **Faire**
Calculer $\mathbf{P}(s_{i-1} = u, s_i = v \,|\, y_1^t, \boldsymbol{\theta})$ puis $\mathbf{P}(s_{i-1} = u \,|\, y_1^t, \boldsymbol{\theta})$ à l'aide de (2.21) et (2.22)
Fin Pour
Fin Pour

La quantité à maximiser devient donc :

$$\begin{aligned}\sum_{s_1^t \in \mathcal{S}^t} \log \mathbf{P}(y_1^t, s_1^t \,|\, \theta) \mathbf{P}(s_1^t \,|\, y_1^t, \theta^{(m)}) = &\sum_{u \in \mathcal{S}} \mathbf{P}(s_1 = u \,|\, y_1^t, \boldsymbol{\theta}) \log \alpha(u) \\ &+ \sum_{u \in \mathcal{S}} \sum_{a \in \Sigma} \mathbf{P}(s_1 = u \,|\, y_1^t, \boldsymbol{\theta}) \log \mu_u(a) \\ &+ \sum_{i=2}^{t} \sum_{u,v \in \mathcal{S}} \mathbf{P}(s_{i-1} = u, s_i = v \,|\, y_1^t, \boldsymbol{\theta}) \log \alpha(u, v) \\ &+ \sum_{i=2}^{t} \sum_{u \in \mathcal{S}} \sum_{a,b \in \Sigma} \mathbf{P}(s_i = u \,|\, y_1^t, \boldsymbol{\theta}) \mathbb{I}_{\{y_{i-1}=a, y_i=b\}} \log \Pi_u(a, b)\end{aligned}$$

Et on en déduit les estimateurs de $\boldsymbol{\theta}^{(m+1)}$ (en adaptant 2.1.2.1) :

$$\alpha^{(m+1)}(u) = \frac{1}{t} \sum_{i=1}^{t} \mathbf{P}(s_i = u \,|\, y_1^t, \boldsymbol{\theta}^{(m)})$$

$$\alpha^{(m+1)}(u, v) = \frac{\sum_{i=2}^{t} \mathbf{P}(s_{i-1} = u, s_i = v \,|\, y_1^t, \boldsymbol{\theta}^{(m)})}{\sum_{i=2}^{t} \mathbf{P}(s_{i-1} = u \,|\, y_1^t, \boldsymbol{\theta}^{(m)})}$$

$$\mu_u^{(m+1)}(a) = \frac{\sum_{i=1}^{t} \mathbb{I}_{\{y_i=a\}} \mathbf{P}(s_i = u \mid y_1^t, \boldsymbol{\theta}^{(m)})}{\sum_{i=1}^{t} \mathbf{P}(s_i = u \mid y_1^t, \boldsymbol{\theta}^{(m)})}$$

$$\Pi_u^{(m+1)}(a,b) = \frac{\sum_{i=2}^{t} \mathbb{I}_{\{y_{i-1}=a, y_i=b\}} \mathbf{P}(s_i = u \mid y_1^t, \boldsymbol{\theta}^{(m)})}{\sum_{i=2}^{t} \mathbb{I}_{\{y_{i-1}=a\}} \mathbf{P}(s_i = u \mid y_1^t, \boldsymbol{\theta}^{(m)})}$$

Ces valeurs correspondent aux estimateurs obtenus par maximum de vraisemblance sur données complètes, après remplacement des états cachés par leur probabilité d'apparition sous $\boldsymbol{\theta}^{(m)}$.

Sous des hypothèses de régularité suffisantes, et pour peu que le paramètre $\boldsymbol{\theta}^{(0)}$ soit dans le voisinage de $\boldsymbol{\theta}^*$, la suite des $\boldsymbol{\theta}^{(m)}$ converge vers le maximum de la vraisemblance (voir par exemple [Mur97]) et la variance des $\theta^{(m)}$ est asymptotiquement la même que celle de $\hat{\theta}$. Si cette hypothèse de proximité n'est pas satisfaite, l'algorithme converge vers un maximum local de la vraisemblance.

La convergence de la suite des paramètres générés par EM est établie dès que les accroissements de la vraisemblance deviennent négligeables. On se fixe donc un critère d'arrêt ε et on considère que la suite de paramètres a convergée dès que $\mathcal{L}(y_1^t \mid \boldsymbol{\theta}^{(m+1)}) - \mathcal{L}(y_1^t \mid \boldsymbol{\theta}^{(m)}) < \varepsilon$.

De plus, θ^* n'étant en général pas connu, on propose généralement plusieurs points de départ tirés aléatoirement ou guidés par un a priori biologique pour $\boldsymbol{\theta}^{(0)}$. On sélectionne ensuite le point limite possédant la vraisemblance maximale.

2.2.2.3 Reconstruction de la suite des états cachés

Une fois les paramètres $\boldsymbol{\theta}^{(\lim)}$ du modèle estimé, on veut généralement "compléter" les données de la séquence en associant à chaque position un état caché. On peut ainsi accéder à deux quantités aux significations légèrement différentes :

a . Le chemin sur les états cachés de probabilité maximale :

$$\overline{s}_1^t = \arg\max_{s_1^t \in \mathcal{S}^t} \left(\mathbf{P}(S_1^t = s_1^t \mid y_1^t, \boldsymbol{\theta}^{(\lim)}) \right)$$

b . La probabilité a posteriori $\mathbf{P}(S_i = u \mid y_1^t, \boldsymbol{\theta}^{(\lim)})$ pour chaque état caché u en toute position i dans la séquence.

$\overline{s}_1^t$ se calcule à l'aide de l'algorithme de Viterbi, qui, en faisant intervenir les équations de récurrence (2.19), (2.20) et (2.21) permet de résoudre ce problème de maximisation en avec une approche par programmation dynamique. Le chemin de probabilité maximale s'obtient donc avec une complexité en $o(t \cdot |\mathcal{S}|)$ (pour plus de détails sur le sujet, regarder par exemple [Rab89]). Notons que le calcul de cette quantité fait intervenir des probabilités qui peuvent être très faibles et impose donc généralement des calculs en échelle logarithmique. Toutefois, $\overline{s}_1^t$ n'est pas toujours le meilleur indicateur pour la reconstruction de la suite des états cachés. En effet, le voisinage de $\mathbf{P}\left(\overline{s}_1^t \mid y_1^t, \boldsymbol{\theta}^{(\lim)}\right)$

peut être très “aplati”, *i.e.* le chemin de probabilité maximale n’est pas particulièrement différent des chemins sous-optimaux. En outre, aucun indice ne permet de statuer sur la significativité du chemin reconstruit.

Le calcul de la probabilité a posteriori pour les états à chaque position permet de résoudre les limitations énoncées pour la quantité (a) en permettant d’obtenir une annotation ainsi qu’un indice de confiance sur cette annotation en chaque position de la séquence. On calcule cette quantité en relançant une itération de l’algorithme *forward-backward*. Remarquons cependant que si des contraintes structurelles sont imposées sur la matrice de transition des états cachés (certaines transitions $\alpha(u,v)$ sont nulles), la reconstruction de la suite des états cachés à l’aide des probabilités a posteriori maximales pourra générer des données complètes de vraisemblance nulle.

2.2.2.4 Score de Fisher pour une CMC

Nous donnons ici les formules permettant de calculer les scores de Fisher pour une séquence, les dérivées de la vraisemblance d’une séquence par rapport aux paramètres. Comme vu précédemment (thèoreme 2.19), ces valeurs sont asymptotiquement gaussiennes, et permettront l’introduction de mesures de similarité entre séquences biologiques conditionnellement à un modèle markovien.

Proposition 2.20. *Soit une séquence de longueur n, $y = y_1, \ldots, y_n$ générée par une Chaîne de Markov cachée de paramètres $\boldsymbol{\theta} = (\alpha, \{\Pi_v, v \in \mathcal{S}\})$. Alors les dérivées de la vraisemblance $\mathbf{P}(y \mid \boldsymbol{\theta})$ sont données par :*

$$\frac{\partial \mathbf{P}(y \mid \boldsymbol{\theta})}{\partial \alpha(u,v)} = \sum_{t=2}^{n} \mathbf{P}(y_1^{t-1}, s_{t-1} = u \mid \boldsymbol{\theta})\, \Pi_v(y_{(t-r_v)\vee 0}^{t-1}, y_t)\, \mathbf{P}(y_{t+1}^{n} \mid s_t = v, \boldsymbol{\theta}) \qquad \forall\, u, v \in \mathcal{S}^2 \tag{2.23}$$

$$= \sum_{t=2}^{n} \frac{\mathbf{P}(y, s_{t-1} = u, s_t = v \mid \boldsymbol{\theta})}{\alpha(u,v)} \tag{2.24}$$

$$= \mathbf{P}(y \mid \boldsymbol{\theta}) \sum_{t=2}^{n} \frac{\mathbf{P}(s_{t-1} = u, s_t = v \mid y, \boldsymbol{\theta})}{\alpha(u,v)} \tag{2.25}$$

$$\frac{\partial \mathbf{P}(y \mid \boldsymbol{\theta})}{\partial \Pi_v(w,i)} = \sum_{t=r_v+1}^{n} \mathbb{I}_{\{y_{t-r_v}^{t} = w.i\}} \frac{\mathbf{P}(y, s_t = v \mid \boldsymbol{\theta})}{\Pi_v(w,i)} \qquad \forall v \in \mathcal{S}, w \in \Sigma^{r_v}, i \in \Sigma \tag{2.26}$$

$$= \mathbf{P}(y \mid \boldsymbol{\theta}) \sum_{t=r_v+1}^{n} \mathbb{I}_{\{y_{t-r_v}^{t} = w.i\}} \frac{\mathbf{P}(s_t = v \mid \boldsymbol{\theta})}{\Pi_v(w,i)} \tag{2.27}$$

(la notation $(t - r_v) \vee 0$ est rajoutée pour tenir compte de la génération des débuts de séquence, soit les r_v premières positions.) Et, notant $q(u)$ la loi d’arrivée sur la fin de

séquence (soit $q(u) = \mathbf{P}(s_{t+1} = \text{'end'} \mid s_t = u)$*) :*

$$\frac{\partial \mathbf{P}(y \mid \boldsymbol{\theta})}{\partial \alpha(u)} = \Pi_u(y_1)\mathbf{P}(y_2^n, s_1 = u \mid y_1, \boldsymbol{\theta}) \tag{2.28}$$

$$= \frac{\mathbf{P}(y, s_1 = u \mid \boldsymbol{\theta})}{\alpha(u)} \tag{2.29}$$

$$\frac{\partial \mathbf{P}(y \mid \boldsymbol{\theta})}{\partial q(u)} = \frac{\mathbf{P}(y, s_n = u \mid \boldsymbol{\theta})}{q(u)} \tag{2.30}$$

Les deux équations (2.27) et (2.25) se calculent facilement à l'aide de la double récurrence sur les équations prédictive, de filtrage et de lissage. On remarquera que ces valeurs sont toujours renormalisées par la vraisemblance incomplète de la séquence. De plus, les quantités calculées s'interprètent aisément en fonction de la qualité d'ajustement d'une séquence au modèle.

Bibliographie

[AB72] R Ash and R Bishop. Monopoly as a markov process. *Mathematics Magazine*, 45 :26–29, January 1972.

[Azz96] A. Azzalini. *Statistical Inference Based on the likelihood*, volume 68 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, 1996.

[Bil61] P. Billingsley. *Statistical Inference for Markov Processes*. Statistical research monographs. University of Chicago Press, 1961.

[BP66] L. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37 :1554–1563, 1966.

[BT82] G. Blom and D. Thorburn. How many random digits are required until given sequences are obtained ? *J. Appl. Prob.*, 19 :518–531, 1982.

[Cow91] R Cowan. Expected frequency of dna patterns using whittle's formula. *Journal of Applied Probability*, 28 :886–892, 1991.

[Fel68] W. Feller. *Introduction to Probability Theory*, volume I. Wiley, third edition, 1968.

[Fre71] D Freedman. *Markov chains*. Holden-Day series in probability and statistics. Holden Day, 1971.

[KH66] S. Karlin and M. Taylor Howard. *A first course in stochastic processes*. New York : Academic Press, 1966.

[MS00] L. Marsan and M.-F. Sagot. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *J. Computational Biology*, 7 :345–362, 2000.

[Mur97] Florence Muri. *Comparaison d'algorithmes d'identification de chaînes de Markov cachées et application à la détection de régions homogènes dans les séquences d'ADN*. PhD thesis, Université René Descartes, Paris V, october 1997.

[Nue] G. Nuel. Numerical solutions for pattern statistics on markov chains. soumis à Journal of Computational Biology.

[Pet69] T. Petrie. Probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 40(1) :97–115, 1969.

[Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceeding of the IEEE*, 77 :257–286, 1989.

[RD99] S. Robin and J.J. Daudin. Exact distribution of word occurrences in a random sequence of letters. *J. Appl. Prob.*, 36 :179–193, 1999.

[RD00] S. Robin and J.J. Daudin. Exact distribution of the distances between any occurrences of a set of words. *Ann. Inst. Statist. Math.*, 2000.

[RSW00] G. Reinert, S. Schbath, and M. Watermann. Probabilistic and statistical properties of words : An overview. *Journal of Computational Biology*, 7 :1–46, 2000.

[vHACV98] J. van Helden, B. André, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computationnal analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281 :827–842, 1998.

[vHdOPO00] Jacques van Helden, Marcel.li del Olmo, and Jose E. Perez-Ortin. Statistical analysis of yeast genomic downstream sequences reveals putative polyadenylation signals. *Nucl. Acids. Res.*, 28(4) :1000–1010, 2000.

[VHRCV00] J. Van Helden, A. F. Rios, and J. Collado-Vides. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Research*, 28(8) :1808–1818, 2000.

[VMS99] A. Vanet, L. Marsan, and M.-F. Sagot. Promoter sequences and algorithmical methods for identifying them. *Res. Microbiol.*, 150 :779–799, 1999.

Chapitre 3

Classification

Le chapitre 2 présentait les modèles markoviens comme une manière probabiliste de représenter les séquences biologiques. Partant de résultats théoriques connus, des exemples d'applications permettaient ensuite de montrer comment certains problèmes en bioinformatique pouvaient être résolus. Inversement, considérons dans ces quelques pages introductives un exemple simple issu de la bioinformatique et nécessitant l'utilisation des outils de l'apprentissage statistique. Ainsi, la mise en application sur un problème particulier pourra illustrer la problématique de classification générale. Nous reviendrons donc à l'exemple que nous introduisons ci-dessous tout au long du chapitre afin d'illustrer les concepts de classification introduits.

Dans le cadre de l'analyse des séquences biologiques, l'annotation fonctionnelle des protéines constitue une des problématiques clefs. Le terme d'annotation fonctionnelle pour une protéine est relativement vague, mais fait par exemple référence à la classe d'enzyme à laquelle celle-ci appartient. Ceci peut aussi correspondre à un ensemble d'annotations Gene Ontology (dont entre autre le champs du composant cellulaire), le groupe d'interactants de cette dernière, ou encore les modifications post-traductionnelles pouvant s'opérer sur celle-ci.

Actuellement, on estime avoir assigné expérimentalement une annotation de type fonctionnel pour seulement une petite fraction des protéines, et la mise en place d'une étude à grande échelle telle qu'une stratégie par double hybride ou spectrométrie de masse peut s'avérer extrêmement coûteuse en temps et en argent. On voudrait donc, partant de l'information disponible (*i.e.* la séquence de la protéine), pouvoir **prédire** un ensemble de caractéristiques fonctionnelles sur celle ci.

Un exemple : la localisation cellulaire des protéines Présentons un exemple simplifié pour la prédiction du compartiment cellulaire. On considère un jeu de L séquences de protéines $(\wp_1, \wp_2, \ldots, \wp_L)$. Pour chacunes d'elles, on dispose d'annotations $(y_1, y_2, \ldots, y_L)$ sur le compartiment cellulaire dans lequel elles résident ($\forall\, i, y_i \in C$ l'ensemble des compartiments possibles).

On souhaite construire une fonction f qui, à partir d'une protéine $\wp$ quelconque, prédirait le compartiment dans lequel celle-ci est localisée avec la plus grande fiabilité possible. On appelle couramment f le classificateur, ou fonction de classification, et l'ensemble $\{(\wp_1, y_1), (\wp_2, y_2), \ldots, (\wp_L, y_L)\}$ le jeu d'entraînement. Rappelons qu'il s'agit ici d'un exemple simplifié, et que la mise en oeuvre de méthodes de prédiction de la localisation cellulaire sera abordée dans les chapitres 6, 5 et 7.2.

Reste à définir sur quel espace de départ notre fonction f pourrait prendre ses valeurs. Le choix le plus complet et le plus naturel serait alors de travailler sur $\Sigma^* = \cup_{n\geq 1}\Sigma^n$, l'ensemble de toutes les séquences possible. Cependant, pour pouvoir travailler naturellement, cet ensemble doit être muni d'une structure permettant au moins la définition d'une mesure. Des méthodes permettant de travailler directement sur la séquence de caractères seront présentées en section 3.3.4.

Dans la suite, on se limitera plutôt aux deux cas de figures suivant pour l'étude des données :

- On suppose connaître une loi de génération des séquences dans chaque classe qui, après estimation, permettra la classification.
- De chaque séquence peptidique $\wp_i$, on peut extraire un vecteur $\mathbf{x}_i$ de d attributs réels ($\mathbf{x}_i \in \mathbb{R}^d$).

3.1 Notations et problématique générale

Posons d'abord le cadre théorique général. Soit $(\mathcal{X}, \tau, \mu)$ un ensemble mesurable (τ la tribu et μ la mesure sur τ). On appelle généralement $\mathcal{X}$ l'espace des attributs ou l'espace des descripteurs. Soit C un ensemble fini, de cardinal $|C|$ et correspondant aux classes.

Formalisons l'approche de l'apprentissage d'un point de vue statistique. Supposons que les couples (x, y) sont générés sur $\mathcal{X} \times C$ suivant une loi de densité F par rapport à μ, *i.e.* :

$$\forall A \in \tau, y \in C, \mathbf{P}(X \in A, Y = y) = \int_{x\in A} F(x, y)\, \mathrm{d}\mu(x, y)$$

Plus pratiquement, F correspond donc à la loi de génération jointe des attributs et des classes. Le but de la classification est alors de trouver une fonction f "qui fait le moins d'erreur" conditionnellement à F. Cette notion d'erreur dépend généralement du problème envisagé, ainsi on voudra dans certains cas minimiser le nombre d'erreurs sur certaines classes en particulier. Par exemple, concernant la prédiction de la localisation cellulaire d'une protéine, certains biologistes s'intéressant au système respiratoire s'attacheront à avoir le moins d'erreurs possibles dans la prédiction des protéines mitochondriales. Définissons donc une fonction C de coût pour les mauvaises classifications. C est définie de $C \times C$ dans $\mathbb{R}$ telle que $C(i, j) \geq 0$ si $i \neq j$ et égale à 0 sinon. En d'autres termes, $C(i, j)$ décrit le coût de mauvaise classification pour un individu de la

classe i prédit comme appartenant à la classe j (notons que C n'est pas nécessairement symétrique). Ceci va nous permettre de définir le risque d'un classifieur.

Définition 3.1 (Risque). Soit f une fonction de classification ($f : \mathcal{X} \to C$). on appelle le risque du classificateur f associé à la fonction de coût C la quantité notée $R(f, C)$:

$$R(f, C) = \mathbb{E}_F\{C(f(x), y)\} = \int C(f(x), y)F(x, y)\,\mathrm{d}\mu(x, y)$$

l'espérance de $C(f(x), y)$ sous F.

Si on prend la fonction de coût classique qui compte le nombre d'erreurs, *i.e.* telle que $C(i, j) = \mathbb{I}_{\{i\neq j\}}$, on obtient alors l'espérance du nombre d'erreurs faites par f sous F :

$$R(f) = \mathbf{P}(f(X) \neq Y) = \int \mathbb{I}_{\{f(x)\neq y\}}F(x, y)\,\mathrm{d}\mu(x, y)$$

La problématique de la classification se reformule donc comme : trouver une fonction de classification $f_{\min}$ qui réalise le risque minimum parmi un ensemble compact de fonctions possibles $\mathcal{F}$:

$$f_{\min} = \arg\min_{f\in\mathcal{F}} R(f)$$

En d'autres termes, quel est le classificateur séparant au mieux les données ?

Une des premières limitations est qu'en général la densité des données F n'est pas connue, et la minimisation du risque ne peut donc pas se ramener a priori à un simple problème d'optimisation sur une classe de fonctions. En effet, revenons à notre exemple sur la prédiction de la localisation cellulaire des protéines. Malgré l'identification de signaux permettant l'adressage des protéines dans certains compartiments, l'état actuel des connaissances ne permet pas la construction d'un modèle précis décidant du compartiment cellulaire à partir d'informations biologiques tirées de la littérature. Par exemple, la tentative initiée par [NN92] n'est pas encore tout à fait concluante (voir 5.4 pour plus de détails).

De plus, on ne dispose que d'un nombre fini de réalisations $\mathcal{I} = \{(x_1, y_1), \ldots, (x_L, y_L)\}$ sur lequel la fonction doit être apprise. Par conséquent, on utilise à la place du risque $R(f, C)$ une estimation du risque :

$$\hat{R}_L(f, C) = \frac{1}{L}\sum_{i=1}^{L} C(f(x_i), y_i)$$

appelé aussi risque empirique. Sous l'hypothèse d'indépendance sur les (x_i, y_i), cet estimateur n'est pas biaisé et converge vers $R(f, C)$ en vertu de la loi des grands nombres. $\mathcal{I}$ est appelé l'échantillon d'entraînement, ou d'apprentissage. On peut alors décider de minimiser le risque empirique :

$$\hat{f}_L^* = \arg\min_{f\in\mathcal{F}} \hat{R}_L(f)$$

Si le risque empirique est estimé sur le jeu de données ayant servi au calcul de $\hat{f}_L^*$, on parle de l'estimateur du risque par resubstitution, qu'on note $\hat{R}_L^d(\hat{f}_L^*)$. Remarquons que cet estimateur est par contre trop optimiste, car si la classe de fonctions $\mathcal{F}$ n'est pas suffisament contrainte, il existe une infinité de fonctions telles que $f(x_i) = y_i$ pour tout i et qui posséderait alors un risque empirique nul. Ce point sera abordé plus rigoureusement dans la section suivante.

Pour prendre en compte ce problème, on utilise donc fréquemment l'estimateur du risque par validation croisée (noté R^{CV} pour "cross-validation"). Pour cela, on découpe le jeu de données en N sous ensembles de taille égale $\mathcal{T}_i = [L/N]$ (en rajoutant les $L\%N$ individus restant pour $\mathcal{T}_N$). On note alors le risque obtenu par validation croisée sur $\mathcal{T}_i$:

$$\hat{R}^{CV}(\hat{f}_i, \mathcal{T}_i) = \frac{1}{|\mathcal{T}_i|} \sum_{(x_j,y_j)\in\mathcal{T}_i} C(\hat{f}_i(x_j), y_j) \quad \text{et} \quad \hat{R}_L^{CV}\left(\left\{\hat{f}_i\right\}_{i=1}^N\right) = \frac{1}{N}\sum_{j=1}^N \hat{R}^{CV}(\hat{f}_i, \mathcal{T}_i)$$

où $\hat{f}_i$ correspond au minimiseur du risque empirique estimé sur les $\mathcal{I} - \mathcal{T}_i$ individus restants.

Si L vaut N, on parle alors du risque "leave one out" (ou LOO). Cet estimateur du risque possède la propriété d'être "presque" non biaisé. Plus précisément, le risque LOO sur m individus est un estimateur non biaisé du risque sur $m-1$ individus :

$$\mathbb{E}_{\mathbf{Z}_{m-1}}[R(f_{\mathbf{Z}_{m-1}})] = \mathbb{E}_{\mathbf{Z}_m}\left[R^{LOO}(f_{\mathbf{Z}_m})\right]$$

En pratique, l'obtention de bornes sur le risque LOO est relativement facile, et ne nécessite pas nécessairement L apprentissages du classificateur. En particulier, une borne sera donnée en section 3.3.1 qui pourra être utilisée ensuite pour la détermination des hyperparamètres d'un classificateur à base de SVM (section 3.3.5). Remarquons cependant qu'on ne dispose d'aucune borne sur la variance du risque estimé par LOO.

3.2 Risque Bayésien

Afin de mieux appréhender le caractère fondamentalement géométrique de l'apprentissage statistique, on supposera dorénavant, sauf mention contraire, que l'espace des attributs $\mathcal{X}$ est un sous ensemble de $\mathbb{R}^d$. Ainsi, on dispose explicitement d'une norme (la norme euclidienne) permettant la comparaison des individus. De plus, afin de simplifier les notations, on ne traitera pour l'instant que les problèmes de classification binaire (en posant $C = \{-1; +1\}$) avec la fonction de coût comptant le nombre d'erreurs indépendamment de la classe (*i.e.* $C(i, j) = \mathbb{I}_{\{i\neq j\}}$). Ceci permet entre autre d'écrire le risque empirique dans une forme plus praticable :

$$\hat{R}_L(f) = \sum_{i=1}^L \frac{1}{2}|f(x_i) - y_i|$$

Nous verrons par la suite comment la méthode par noyaux peut permettre de s'affranchir de cette limitation sur l'espace des attributs (sections 3.3.3 et 3.3.4).

Supposons maintenant que la distribution F sous-tendant la génération des données est connue. On appelle alors le classificateur qui minimise le risque connaissant cette distribution le classificateur de Bayes, noté f^*. Le risque associé $R(f^*)$ est appelé le risque de Bayes et noté R^*. Le classificateur de Bayes est un des classificateurs de risque minimum et correspond donc à l'erreur minimum attendue.

Si on connaît la distribution des données, le classificateur de Bayes s'obtient avec la probabilité a posteriori de la classe conditionnellement aux données. Soit :

$$f^*(x) = \begin{cases} 1 & \text{si } \mathbf{P}(Y = 1 \mid X = x) > \frac{1}{2} \\ -1 & \text{sinon} \end{cases}$$

où la loi a posteriori s'obtient à partir de la formule de Bayes :

$$\mathbf{P}(Y = 1 \mid X = x) = \frac{\mathbf{P}(X = x \mid Y = 1)\mathbf{P}(Y = 1)}{\mathbf{P}(X = x \mid Y = -1)\mathbf{P}(Y = -1) + \mathbf{P}(X = x \mid Y = 1)\mathbf{P}(Y = 1)}$$

L'analyse discriminante propose de résoudre ce problème de minimisation du risque en estimant les lois des observations conditionnelles aux classes ($F_y = \mathbf{P}(X \mid Y = y)$) et les lois a priori pour chaque classe ($\pi_i = \mathbf{P}(Y = i)$).

On peut alors au choix :

- Estimer les densités $\hat{F}_y$ conditionnelles aux classes et les proportions des classes $\hat{\pi}(y)$ avec des méthodes non paramétriques, en faisant par exemple de l'estimation de densité par des fonctions noyaux.
- Supposer que les F_y sont tirées suivant une distribution paramétrique connue. Par exemple l'analyse discriminante de Fisher suppose que les F_y suivent une loi normale. Cette loi normale est en général préférée car : (1) elle se retrouve dans un grand nombre de phénomènes naturels, (2) l'analyse discriminante à la Fisher est dans la pratique très robuste, même en présence de données non gaussiennes.

Dans les deux cas, on substitue alors à f^* le classificateur $\hat{f}^*$ construit comme suit :

$$\hat{f}^*(\mathbf{x}) = \begin{cases} 1 & \text{si } \dfrac{\hat{f}_1(\mathbf{x})\hat{\pi}_1}{\hat{f}_{-1}(\mathbf{x})\hat{\pi}_{-1} + \hat{f}_1(\mathbf{x})\hat{\pi}_1} > \dfrac{1}{2} \\ -1 & \text{sinon} \end{cases}$$

Revenons à nouveau à notre exemple. Dans le cadre proposé, il est maintenant nécessaire d'extraire de chaque protéine un vecteur d'attributs à valeurs réelles. Une solution proposée couramment consiste alors à utiliser comme descripteurs les fréquences des acides aminés ou des peptides de k lettres présents dans la séquence. Ainsi on replonge implicitement chaque protéine $\wp_i$ dans $\mathbb{R}^d$ (avec $d = |\Sigma|^k$) à l'aide de la fonction

Φ définie comme suit :

$$\Phi : \Sigma^* \to \mathbb{R}^d$$
$$\wp_i \mapsto \mathbf{x}_i = \left(\frac{N_a(\wp_i)}{N_\bullet(\wp_i)}\right)_{a\in\Sigma^k}$$

(où $N_a(\wp_i)$ est le nombre d'occurrences de a dans wp_i et $N_\bullet(\wp_i)$ la longueur de $\wp_i$). Bien entendu, une séquence protéique n'est pas complètement décrite par les fréquences de ses mots de k lettres. D'autres descripteurs à valeur réelle peuvent être proposés et seront présentés pour leurs applications à la prédiction du compartiment cellulaire dans la section 5.2. En outre, les exemples seront maintenant traités pour deux compartiments, les protéines localisées dans le cytoplasme et celles sécrétées à l'extérieur de la cellule.

Remarque modèles markoviens : Remarquons que pour l'analyse discriminante, il est simplement nécessaire de pouvoir se donner une loi sur $\mathcal{X}$ conditionnelle à la classe pour permettre la classification. Par exemple si la classification fait intervenir les séquences biologiques, il n'est alors pas obligatoirement nécessaire de replonger les séquences dans $\mathbb{R}^d$. En effet en se donnant une loi de génération des séquences conditionnellement au compartiment, les méthodes de l'analyse discriminante s'appliquent. Au vu des considérations du chapitre précédent, on peut par exemple supposer que les séquences du compartiment i sont générées suivant une chaîne de Markov de matrice de transition Π_i et de distribution stationnaire μ_i (déduite de Π_i) :

$$f_i(\wp) = L(\wp; \Pi_i, \mu_i)$$

et le classificateur peut s'écrire comme un rapport de vraisemblance (équivalent à la forme donnée plus haut) :

$$\hat{f}^*(\wp) = \begin{cases} 1 & \text{si } \dfrac{\hat{\pi}_1}{\hat{\pi}_{-1}} \cdot \dfrac{L(\wp; \hat{\Pi}_1, \hat{\mu}_1)}{L(\wp; \hat{\Pi}_{-1}, \hat{\mu}_{-1})} > 1 \\ -1 & \text{sinon} \end{cases}$$

Où les $\hat{\Pi}_i$ ont été estimés par maximum de vraisemblance sur les protéines du compartiment i. Remarquons que l'hypothèse markovienne pour une séquence protéique est extrêmement contraignante. En particulier, l'hypothèse d'homogénéité le long de la séquence n'est pas satisfaite.

Analyse discriminante : Détaillons maintenant le cas de l'analyse discriminante à la Fisher. On suppose que les $\mathbf{x}_i$ apppartenant à la classe i sont générés suivant une

gaussienne $\mathcal{N}(\mathbf{g}_i, \mathcal{V}_i)$. La classification se fait alors par :

$$\forall i \in C$$

$$\text{estimer : } \mathbf{g}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{x}_j \qquad \text{(moyennes)}$$

$$\mathcal{V}_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (\mathbf{x}_j - \mathbf{g}_i) \,.\, (\mathbf{x}_j - \mathbf{g}_i)^\top \qquad \text{(variances-covariances)}$$

Et $\hat{f}^*$ s'obtient par :

$$\hat{f}^* = \arg\min_{i \in C} \left\{ \hat{\pi}_i \|\mathbf{x} - \mathbf{g}_i\|_{\mathcal{V}_i} \right\} = \arg\min_{i \in C} \left\{ \hat{\pi}_i . (\mathbf{x} - \mathbf{g}_i) \mathcal{V}_i^{-1} (\mathbf{x} - \mathbf{g}_i)^\top \right\}$$

l'extraction des attributs impose un modèle : Revenons à l'exemple sur les modèles markoviens afin d'examiner les liens possibles avec l'analyse discriminante. Afin de simplifier le problème, supposons qu'un modèle d'ordre 0 est estimé sur des séquences toutes de même longueur ℓ. Alors, si $\hat{\boldsymbol{\theta}}^i$ représente le vecteur estimé par maximum de vraisemblance sur toutes les lettres pour le compartiment i, on a :

$$\forall a \in \Sigma \quad \hat{\theta}^i_a = \frac{1}{N_i} \sum_{j \,|\, \wp_j \in i} \frac{N_a(\wp_j)}{\ell}$$

on a alors $\hat{\boldsymbol{\theta}}^i = \mathbf{g}_i$. Comme montré précédemment (sections 2.1.2.2 et 2.19) les estimateurs par maximum de vraisemblance pour une CM sont asymptotiquement gaussiens.

Ainsi, la classification discriminante quadratique sur les fréquences des acides aminés $\mathbf{p}^i_j$ équivaut aymptotiquement (sur ℓ) à une classification sur les paramètres estimés $\hat{\boldsymbol{\theta}}^i_j$, après avoir supposé les séquences générées par un modèle M_0. En d'autres termes, on estime $\sqrt{l}^{-1} \mathcal{J}(\theta_i^\star)^{-1}$ par les variances observées conditionnellement aux classes $\mathcal{V}_i$ et l'hypothèse markovienne justifie l'utilisation de l'analyse discriminante quadratique. Cette comparaison s'étend directement aux relations entre modèles markoviens d'ordre k et l'utilisation des fréquences des mots de $k+1$ lettres.

Remarquons que la démarche de l'analyse discriminante nous éloigne du but initial. En effet, on substitue à la recherche d'un classificateur de risque minimum égal au risque bayésien, une estimation du classificateur résolvant le risque bayésien. En d'autres termes, au lieu de minimiser directement le risque du classificateur, on minimise le risque conditionnellement à une loi supposée sur les densités des classes.

Quelques bornes pour la minimisation du risque empirique Afin d'introduire le point de vue motivant l'utilisation des SVM, étudions la validité asymptotique de la

stratégie de minimisation du risque empirique. A quelles conditions peut-on dire que l'on converge vers le classificateur de Bayes par minimisation du risque empirique ?

Le fléau de la dimensionnalité[1] impose un nombre minimum d'exemples augmentant exponentiellement suivant la dimensionnalité du problème. L'approche proposée par Vapnik et Chervonenkis propose de s'affranchir de ce point de vue en ne considérant la dimensionnalité du problème qu'à travers le pouvoir séparateur maximal permis par la classe de fonctions considérées.

Remarquons que, en vertu de la loi faible des grand nombres, on a :

$$\forall f \text{ continue bornée} : \quad \hat{R}_L(f) \xrightarrow[L\to\infty]{} R(f)$$

Plus précisement, l'inégalité de Hoeffding donne une convergence en probabilité exponentielle pour le risque empirique d'une fonction f fixée :

$$\forall \varepsilon > 0,\ \mathbf{P}\Big(\big|\hat{R}_L(f) - R(f)\big| > \varepsilon\Big) \leq 2\exp^{-2L\varepsilon^2}$$

Cependant, cela ne donne pas un majorant sur le minimum du risque espéré. En d'autre termes, si on a deux classificateurs f_1 et f_2 tels que $\hat{R}(f_1) < \hat{R}(f_2)$, on ne peut pas conclure que $R(f_1) < R(f_2)$.

Par contre, en démontrant la convergence uniforme du risque empirique d'un classificateur vers le vrai risque, on peut obtenir ce résultat. Le but est donc d'obtenir une borne exponentielle sur $\mathbf{P}\Big\{\sup_{f\in\mathcal{F}} \big|\hat{R}_L(f) - R(f)\big| > \varepsilon\Big\}$. L'approche de Vapnik-Chervonenkis permet de reformuler cette borne en fonction du pouvoir séparateur de la classe de fonctions $\mathcal{F}$ considérée.

Posons quelques définitions permettant de travailler avec la "dimension" associée au pouvoir séparateur de $\mathcal{F}$ avant d'énoncer la borne obtenue.

Définition 3.2 (shatter-coefficient). Soit $\mathcal{F}$ une classe de classificateurs binaires de $\mathbb{R}^d$ dans $\{-1; +1\}$. On définit le **shatter-coefficient** (ou coefficient de hachage) pour m points associé à $\mathcal{F}$, $\mathcal{S}(\mathcal{F}, m)$ comme :

$$\mathcal{S}(\mathcal{F}, m) = \max_{(x_1,\ldots,x_m)\in\mathbb{R}^d} \underbrace{Card\Big\{(f(x_1), \ldots, f(x_m));\ f \in \mathcal{F}\Big\}}_{\text{cardinal de } \mathcal{F} \text{ restreint à } (x_1,\ldots,x_m)}$$

En d'autres termes, $\mathcal{S}(\mathcal{F}, m)$ mesure le nombre de manières suivant lesquelles une fonction de $\mathcal{F}$ peut au plus séparer m points. Par exemple, si $\mathcal{S}(\mathcal{F}, m) = 2^m$, alors il existe m points pouvant être séparés de toutes les manières possibles par les fonctions de $\mathcal{F}$.

Définition 3.3 (VC-dimension). Soit $\mathcal{F}$ une classe de classificateurs binaires de $\mathbb{R}^d$ dans $\{-1; +1\}$. On appelle dimension de Vapnik-Chervonenkis (ou VC-dimension) le plus grand entier $k \geq 1$ pour lequel $\mathcal{S}(\mathcal{F}, k) = 2^k$ et on le note $\mathcal{V}_C$. Si $\forall\, n, \mathcal{S}(\mathcal{F}, n) = 2^n$ on pose $\mathcal{V}_C = \infty$

[1] "curse of dimensionnality", une expression populaire due à Bellman

La VC-dimension d'un groupe de classificateurs permet donc de statuer sur le nombre maximum de points séparables en 2 groupes. Bien sûr, la VC-dimension est reliée au shatter coefficient par la relation suivante : $\mathcal{S}(\mathcal{F}, n) \leq n^{\mathcal{V}_C}$

Alors, en adaptant la démonstration du théorème de Glivenko-Cantelli (qui donne une vitesse de convergence pour la mesure empirique sur les fonctions de répartition), on obtient la borne suivante, ou inégalité de Vapnik-Chervonenkis :

Théorème 3.4 (inégalité de Vapnik-Chervonenkis). *Soit $\mathcal{F}$ une classe de classificateurs, on a :*

$$\mathbf{P}\left\{\sup_{f\in\mathcal{F}} \left|\hat{R}_n(f) - R(f)\right| > \varepsilon\right\} \leq 8\mathcal{S}(\mathcal{F}, n) \exp^{-n\varepsilon^2/32} \tag{3.1}$$

Grâce à cette propriété de convergence uniforme, on peut ainsi montrer que, si la VC-dimension de $\mathcal{F}$ est finie, $\hat{R}(\hat{f}_L^*)$ converge bien vers $R(f_{\mathcal{F}}^*)$. Donc si la classe $\mathcal{F}$ contient bien le classificateur de Bayes, la stratégie de minimisation du risque empirique converge vers le minimum du risque sur la classe de classificateurs, pour peu que la VC-dimension de $\mathcal{F}$ soit finie.

Ainsi, dans le cas où la règle de Bayes appartient à la classe des classificateurs, on obtient bien une convergence pour notre minimiseur du risque empirique. Cependant, pour des problèmes réels, la règle de Bayes n'appartient en général pas à la classe de classificateurs considérés. Si la classe de fonctions est plus grande, il se peut aussi que le nombre de paramètres nécessaires à l'estimation soit trop élevé par rapport au nombre d'exemples pris en compte.

L'approche par minimisation du risque structurel, utilisant le formalisme présenté rapidement dans les pages précédentes, propose une approche alternative du problème, dont une des applications peut être les SVM.

3.3 Support Vector Machine

Cette section présente la méthodologie par "Support Vector Machines" (SVM) et quelques unes des adaptations proposées dans la littérature pour des problèmes de classification sur les séquences biologiques.

Initiée par Vapnik, cette méthode de classification combine deux idées simples pour produire un classificateur réalisant de bonnes performances. Le premier point résout explicitement le problème de minimisation d'un critère de risque dans le cas où la classe des fonctions séparatrices est limitée aux hyperplans de $\mathbb{R}^d$. Ensuite, l'utilisation de fonctions noyaux (ou kernels), par un surdimensionnement implicite de l'espace des attributs, permet l'utilisation de fonctions séparatrices plus complexes, tout en conservant les avantages tant théoriques que calculatoires liés à la problématique de détermination d'un hyperplan.

Dans la suite, nous présenterons les différentes étapes permettant la mise en place d'un classificateur par SVM. Les premiers paragraphes abordent volontairement le problème simple, permettant ainsi de dégager les quantités d'intérêt lors de la détermination de la solution. Le contenu des sections est donc en substance le suivant :

section 3.3.1 : Présentation de la problématique de minimisation du risque structurel, qui, avec l'approche VC, permet d'obtenir des bornes à horizon fini sur le risque d'une classe de fonctions séparatrices. En reformulant cette borne dans le cas de fonctions de séparation limitées aux hyperplans, le risque est minimisé uniformément en identifiant l'hyperplan de marge maximale.

section 3.3.2 : Reformulation du problème de détermination de l'hyperplan de marge maximale dans le cas de données séparables puis non séparables. Ceci nous ramène à un problème d'optimisation convexe sous contrainte. L'hyperplan séparateur est alors déterminé comme une combinaison linéaire des exemples "les plus proches" de la frontière (les Support Vectors).

section 3.3.3 : Introduction des kernels, ou fonctions à noyaux, qui définissent implicitement un produit scalaire sur un espace hilbertien[2]. Le problème d'optimisation présenté à la section précédente reste inchangé, et permet ainsi la construction de frontières de décision adaptées aux problèmes réels.

Une fois la méthodologie générale présentée, certaines questions nécessitent une attention particulière lors de la mise en oeuvre pratique. La première est la détermination des hyperparamètres optimaux associés au noyau utilisé (section 3.3.5).

En outre, les fonctions à noyau permettent de définir une mesure de similarité entre éléments d'un ensemble quelconque. On présentera quelques noyaux qui ont été proposés pour permettre l'étude des séquences biologiques (section 3.3.4), fondées directement sur les mots présents dans la séquence (3.3.4.1), ou utilisant un a priori à travers un modèle probabiliste sur les séquences (3.3.4.2).

3.3.1 Minimisation du risque structurel

Revenons à la borne présentée dans le théorème 3.4 :

$$\mathbf{P}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f)-R(f)\right|>\varepsilon\right\}\leq 8\mathcal{S}(\mathcal{F},n)\exp^{-n\varepsilon^2/32}$$

En fixant le terme de gauche à δ, on peut résoudre l'équation en ε, soit :

$$\delta \leq 8\mathcal{S}(\mathcal{F},n)\exp^{-n\varepsilon^2/32}$$

$$\text{ou} \quad \varepsilon \leq \sqrt{\frac{32}{n}\left(\log\left(\mathcal{S}(\mathcal{F},n)\right)+\log\frac{8}{\delta}\right)}$$

[2] un espace hilbertien est un espace vectoriel pouvant être muni d'un produit scalaire introduisant une norme et une distance

Alors, avec une probabilité $(1 - \delta)$, on a :

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{32}{n}\left(\log\left(\mathcal{S}(\mathcal{F}, n)\right) + \log\frac{8}{\delta}\right)} \tag{3.2}$$

Remarquons que la VC-dimension de $\mathcal{S}(\mathcal{F}, n)$ permet d'obtenir une borne explicite en n sur cette dernière quantité :

$$\mathcal{S}(\mathcal{F}, n) \leq \mathcal{V}_{\mathcal{F}} \log\left(\frac{n}{\mathcal{V}_{\mathcal{F}}}\right) + 1$$

La stratégie de minisation du risque empirique ne travaille alors que sur la première quantité, pour une classe $\mathcal{F}$ fixée. Vapnik propose une stratégie de minimisation du risque structurel. En se donnant une suite de fonctions de classifications $\mathcal{F}_i$ de VC-dimension croissante, on va choisir le classificateur qui réalise le meilleur compromis entre la minimisation du risque empirique et du second terme, afin d'obtenir une borne uniforme sur n et sur les $\mathcal{F}_i$.

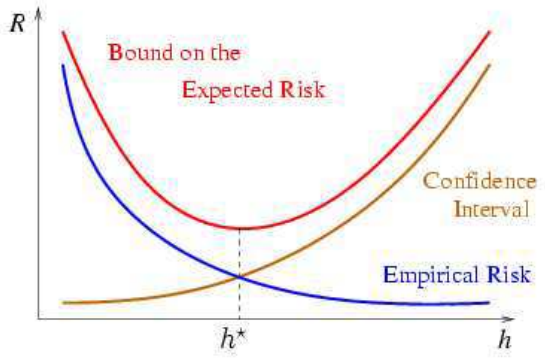


FIG. 3.1 – diagramme schématique des différentes bornes en fonction de la capacité du classificateur, à taille d'ensemble d'entraînement fixé.

Limitons la classe de fonctions séparatrices aux hyperplans séparateurs de $\mathbb{R}^d$. Avant de voir de quelle manière les bornes sur le risque peuvent s'exprimer dans ce cas, définissons quelques valeurs d'intérêt pour l'étude des classificateurs à marge.

La fonction de décision f définie à partir d'un hyperplan peut s'écrire :

$$f(x) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + \mathbf{b}\right)$$

où $\mathbf{w}$ est un vecteur normal à l'hyperplan, et $\mathbf{b}$ le décalage à l'origine. f sépare alors l'espace en deux classes suivant son signe, *i.e.* le signe de $\langle \mathbf{w}, \mathbf{x} \rangle + \mathbf{b}$.

On définit alors la marge $\gamma_{\mathbf{w},\mathbf{b}}$, qui correspond à la distance du plus proche des points à l'hyperplan :

$$\gamma_{\mathbf{w},\mathbf{b}} = \min_{i=1,\dots,n} (|\langle \mathbf{w}, \mathbf{x}_i \rangle + \mathbf{b}|)$$

La marge peut aussi être vue comme le minimum des distances des enveloppes convexes des groupes de points de part et d'autre de l'hyperplan.

Cependant, cette marge dépend des valeurs de $\mathbf{w}$ et $\mathbf{b}$, *i.e.* $\gamma_{\lambda\mathbf{w},\lambda\mathbf{b}} = \lambda\gamma_{\mathbf{w},\mathbf{b}}$. Afin d'éviter toute ambiguité on travaille avec la forme canonique de l'hyperplan, c'est à dire les valeurs de $\mathbf{w}$ et $\mathbf{b}$ pour lesquelles $\gamma_{w,b}$ vaut 1. La longueur de la marge (ou marge géométrique) peut alors directement s'exprimer en fonction de $\mathbf{w}$:

$$\gamma_{\text{geom}} = \frac{1}{\|\mathbf{w}\|_2}$$

L'hyperplan étant implicitement pris "au milieu" des exemples.

Commençons par un théorème qui relie directement la marge d'un hyperplan de $\mathbb{R}^d$ à son shatter-coefficient. Ceci permettra la construction d'une borne sur le risque de ces classificateurs de la forme (3.2) qui justifie ainsi l'approche par SVM. Les deux théorèmes qui suivent sont directement tirés de [SS01], chapitres 5 et 7.

Théorème 3.5. *Considérons les hyperplans du type* $\langle \mathbf{w}, \mathbf{x} \rangle = 0$ *avec* $\mathbf{w}$ *normalisé de telle manière que pour un jeu de points* $X^* = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r\}$ *on a :*

$$\min_{i=1,\dots,r} |\langle \mathbf{w}, \mathbf{x}_i \rangle| = 1$$

Alors, le jeu de fonctions de décisions $f_{\mathbf{w}}(\mathbf{x})$ *définies sur* X^* *par* $\operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle)$*, et telles que* $\|\mathbf{w}\| \leq \Lambda$ *ont une dimension de Vapnik-Chervonenkis* $\mathcal{V}_{\mathbf{w}}$ *qui satisfait à l'inégalité :*

$$\mathcal{V}_{\mathbf{w}} \leq R^2\Lambda^2$$

où R est le rayon de la plus petite sphère centrée à l'origine et contenant X^**.*

Le théorème est donné ici uniquement pour les hyperplans passant par l'origine, remarquons que par un surdimensionnement adéquat de l'espace des attributs ($\tilde{\mathbf{x}} = (\mathbf{x}, 1)$ et $\tilde{\mathbf{w}} = (\mathbf{w}, b)$), on peut obtenir un résultat équivalent dans le cas général. La figure 3.3 illustre la dépendance du pouvoir séparateur de l'hyperplan suivant la taille de la marge (bornée ici par $\frac{1}{\Lambda}$). Ainsi en fixant le rayon de la boule à laquelle les exemples appartiennent, on voit que la taille de la marge contraint le nombre maximal de points séparables.

Théorème 3.6 (Borne sur les séparateurs à marge). *Soit le jeu de fonctions de décision* $\mathcal{F} = \{f(x) = \langle w, x \rangle, \|w\| \leq \Lambda, \|x\| \leq R\}$*, avec R et* Λ *dans* $\mathbb{R}^+$*. De plus, pour un ensemble d'entraînement de n individus, soient* $\rho > 0$*, et* $\hat{R}_\rho$ *la proportion des exemples*

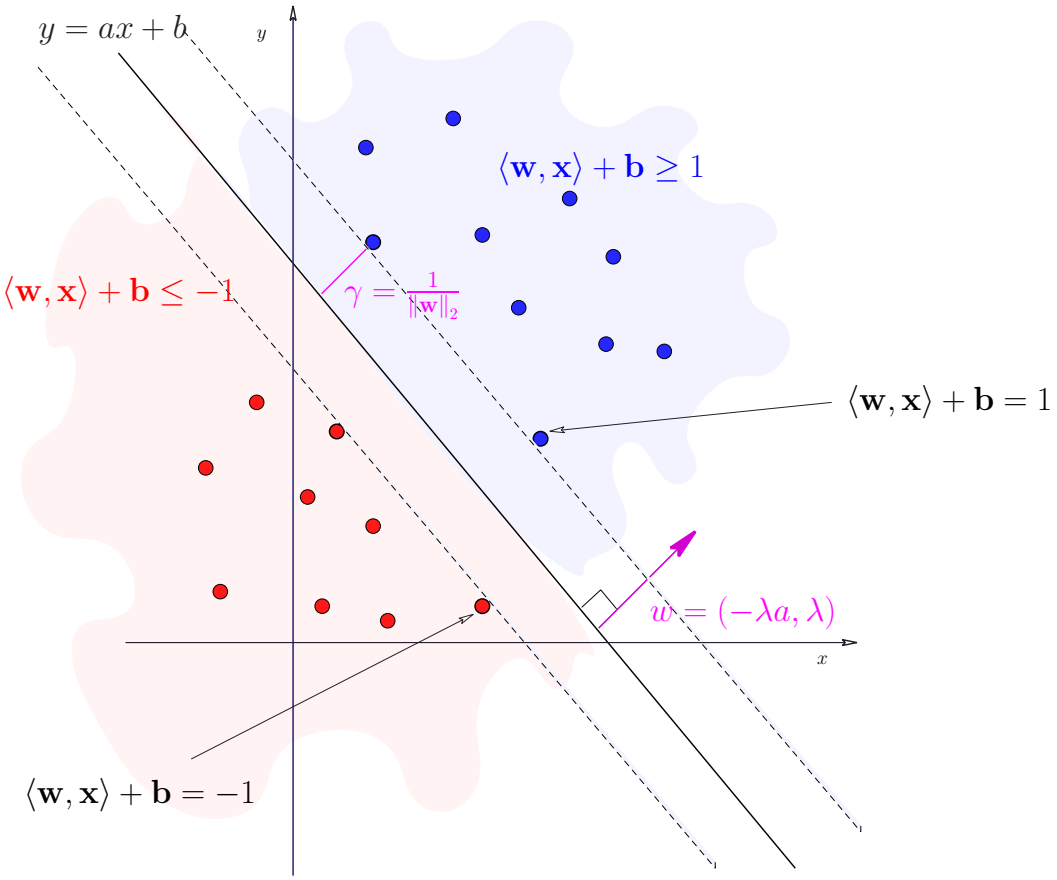


FIG. 3.2 – Représentation dans $\mathbb{R}^2$ des quantités impliquées pour un hyperplan séparateur (ici une droite). Afin d'éviter toute ambiguité le vecteur w est renormalisé, *i.e.* $\langle \mathbf{w}, \mathbf{x} \rangle + \mathbf{b} \geq 1$.

d'entraînement possédant une marge plus petite que $\rho/\|w\|$. *En d'autres termes,* $\hat{R}_\rho$ *correspond au risque empirique associé à la marge de taille* ρ. *Alors, pour toute distribution* F *sur* $\mathcal{X} \times C$ *générant les données, on a, avec une probabilité supérieure à* $1 - \delta$ *:*

$$R(f) < \hat{R}_\rho + \sqrt{\frac{c}{n}\left(\frac{R^2\Lambda^2}{\rho^2}\ln^2 n + \ln\frac{1}{\delta}\right)}$$

où c est une constante.

Cette borne justifie dans une certaine mesure la stratégie par SVM. En effet, la quantité $R^2\Lambda^2$ est en général fixée, et pour minimiser le risque, on va donc essayer de prendre le ρ maximal. Cependant, pour ρ trop grand, c'est la quantité $\hat{R}_\rho$ qui augmente. Cette borne exprime bien le compromis cherché entre une classe de fonctions de capacité

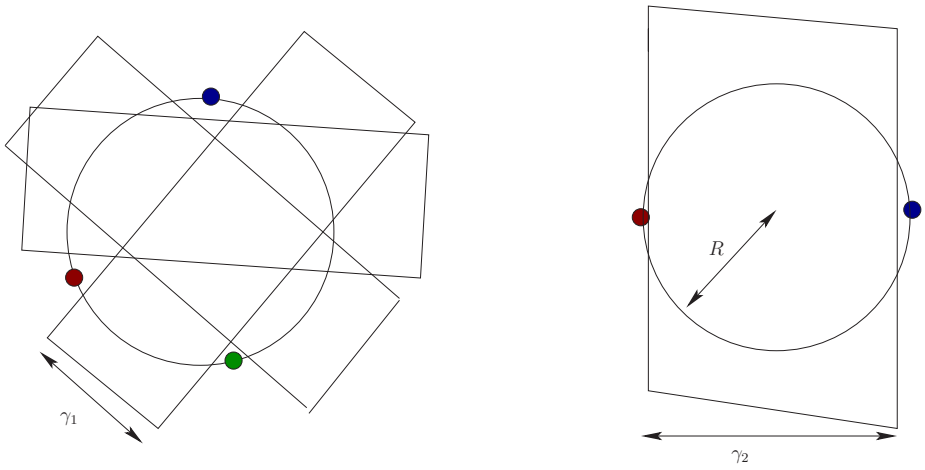


FIG. 3.3 – visualisation de l'influence de la marge sur le pouvoir séparateur d'un hyperplan pour des points dans $\mathbb{R}^2$. Quand γ augmente, l'hyperplan ne peut plus séparer au maximum que deux points.

minimum et la minimisation de l'erreur obtenue sur les exemples d'entraînement. La section suivante présente les détails d'implémentation proposés pour ce problème.

3.3.2 Entraînement des Séparateurs à marge optimale.

Le théorème énoncé à la section précédente met en valeur le but recherché lors de la construction de l'hyperplan séparateur optimal au sens du risque. On souhaite maximiser la marge en restreignant le nombre d'erreurs faites lors de l'entraînement du classificateur. Cette section détaille dans quelle mesure ce problème peut se réécrire, tout d'abord dans le cas séparable, puis en introduisant une fonction de coût pour les exemples mal classés.

Ainsi dans la suite, on considère n individus : $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1; +1\}$. L'**entraînement** du SVM consiste alors en la détermination de la fonction de classification $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + \mathbf{b}$ définie par $(\mathbf{w}, \mathbf{b})$. On verra par la suite que cette fonction peut aussi s'écrire comme une combinaison linéaire des $\mathbf{x}_i$.

3.3.2.1 Cas des classes séparables : SVMs à "marge dure"

Supposons dans un premier temps que les individus sont séparables linéairement, *i.e.* l'enveloppe convexe des individus des deux classes est d'intersection vide. D'après le théorème de Hahn-Banach, on sait qu'il existe au moins un hyperplan $(\mathbf{w}, \mathbf{b})$ tel que $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle + \mathbf{b}) > 0 \quad \forall i = 1, \ldots, n$. On cherche alors l'hyperplan possédant la marge maximale. En imposant un hyperplan canonique, on se ramène au problème d'optimi-

sation sous contrainte suivant :

$$\min_{\mathbf{w},\mathbf{b}} \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle$$
$$\text{soumis à } \; y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + \mathbf{b}) \geqslant 1$$

La fonction à minimiser est strictement convexe, et les contraintes linéaires. Ce problème d'optimisation admet donc une unique solution. Pour déterminer la solution de cette minimisation sous contrainte, étudions le lagrangien du problème :

$$\mathrm{L}(\mathbf{w}, \mathbf{b}, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle - \sum_{i=1}^{n} \alpha_i [y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + \mathbf{b}) - 1] \tag{3.3}$$

La solution du problème correspond alors à un point selle du Lagrangien vérifiant la condition dite de Karun-Kush-Tucker (KKT). En d'autres termes, $\mathbf{w}, \mathbf{b}$ et $\boldsymbol{\alpha}$ vérifient :

$$\begin{aligned}
\delta_{w^j} \mathrm{L}(\mathbf{w}, \mathbf{b}, \boldsymbol{\alpha}) &= 0 \\
&= w^j - \sum_{i=1}^{n} \alpha_i y_i x_i^j && (3.4)\\
\delta_{b^j} \mathrm{L}(\mathbf{w}, \mathbf{b}, \boldsymbol{\alpha}) &= 0 \\
&= \sum_{i=1}^{n} \alpha_i y_i && (3.5)\\
\delta_{\alpha^j} \mathrm{L}(\mathbf{w}, \mathbf{b}, \boldsymbol{\alpha}) \leqslant 0 &\rightleftharpoons y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + \mathbf{b}) - 1 \geqslant 0 && (3.6)\\
\forall i = 1, \ldots, n \quad \alpha_i &\geqslant 0 && (3.7)\\
\alpha_i [y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + \mathbf{b}) - 1] &= 0 \quad \text{(KKT)} && (3.8)
\end{aligned}$$

Plusieurs propriétés intéressantes sur les multiplicateurs de Lagrange ressortent de ces relations :

(3.4) : L'hyperplan séparateur s'écrit comme une combinaison linéaire des individus pondérés par les multiplicateurs de Lagrange α_i. On a donc pour $\mathbf{x} \in \mathbb{R}^d$:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x}\rangle + \mathbf{b} = \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x}\rangle + \mathbf{b}$$

(3.8) : Ces susdits α_i ne sont non nuls que pour les points qui sont sur la marge. Ceci est en accord avec l'idée intuitive que l'hyperplan se construit uniquement avec les points de la marge. On appelle aussi ces points les Support Vectors (d'où le nom de la méthode).

(3.5) : Les pondérations se compensent exactement entre les deux classes.

En réinjectant les relations (3.4,3.5) dans (3.3), on peut réécrire le lagrangien comme une fonctionnelle de $\boldsymbol{\alpha}$:

$$\mathrm{L}(\mathbf{w}, \mathbf{b}, \boldsymbol{\alpha}) = W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

Et le problème peut se réécrire en son équivalent dual (un résultat dû à Wolfe) :

$$\max_{\alpha} W(\alpha)$$
$$\text{soumis à } \sum_{i=1}^{n} \alpha_i y_i = 0$$
$$\forall i = 1, \ldots, n \quad \alpha_i \geqslant 0$$

Remarquons que la fonctionnelle duale ne s'écrit plus qu'en fonction de la matrice des produits scalaires $\left(\langle \mathbf{x}_i, \mathbf{x}_j \rangle\right)_{i,j=1,\ldots,n}$ entre les individus.

3.3.2.2 Permettre des erreurs : SVMs à marge souple

Afin de pouvoir formuler le relâchement sur la contrainte en termes de SVMs, on introduit les variables ξ_i (couramment appelées "slack"[3] variables), qui mesurent l'écartement à la marge pour les individus mal classés :

$$\xi_i = 0 \vee (1 - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b))$$

Ceci permet de réécrire la contrainte sur chaque point comme :

$$\forall i = 1, \ldots, n \quad y_i(\langle w, x_i \rangle + b) \geqslant 1 - \xi_i$$

Par contre, il est nécessaire de pénaliser ces ξ_i, auquel cas des valeurs suffisamment grandes de ξ_i seraient solutions pour n'importe quel hyperplan. Le problème de minimisation se réécrit alors :

$$\min_{\mathbf{w},\mathbf{b},\boldsymbol{\xi}} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + \frac{C}{n} \sum_{i=1}^{n} \xi_i \quad \text{marges souples dans } \mathbb{L}^1 \tag{3.9}$$

$$\min_{\mathbf{w},\mathbf{b},\boldsymbol{\xi}} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + \frac{C}{n} \sum_{i=1}^{n} \xi_i^2 \quad \text{marges souples dans } \mathbb{L}^2 \tag{3.10}$$

soumis aux contraintes

$$\forall i = 1, \ldots, n \quad y_i(\langle w, x_i \rangle + b) \geqslant 1 - \xi_i \tag{3.11}$$
$$\xi_i \geqslant 0 \tag{3.12}$$

[3] littéralement variables de mollesse

C correspond donc au coût de mauvaise classification. Notons que les pénalisations $\mathbb{L}^1$ ou $\mathbb{L}^2$ n'impliquent pas la même fonction de pénalisation pour les exemples mal classés.

Nous ne présentons ici que la formulation du problème à marges souples dans $\mathbb{L}^1$. Comme on le verra par la suite (section 3.3.5), le cas $\mathbb{L}^2$ peut être traité de manière similaire au cas des marges dures après surdimensionnement.

Le lagrangien du problème de minimisation (3.9) s'écrit donc :

$$\mathrm{L}(\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + \frac{C}{n} \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i [y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + \mathbf{b}) - 1 + \xi_i] - \sum_{i=1}^{n} \beta_i \xi_i$$

Et on obtient comme jeu de conditions :

conditions sur les variables :

$$\begin{aligned}
\delta_{w^j} \mathrm{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= 0 \\
&= w^j - \sum_{i=1}^{n} \alpha_i \, y_i \, x_i^j && (3.13) \\
\delta_{b^j} \mathrm{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= 0 \\
&= \sum_{i=1}^{n} \alpha_i \, y_i && (3.14) \\
\delta_{\xi_i} \mathrm{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= 0 \\
&= \frac{C}{n} - \beta_i - \alpha_i && (3.15) \\
\text{(en particulier } \alpha_i &= \frac{C}{n} - \beta_i) && (3.16)
\end{aligned}$$

conditions sur les variables duales :

$$\begin{aligned}
\delta_{\alpha_i} \mathrm{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\leqslant 0 && (3.17) \\
\delta_{\beta_i} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\leqslant 0 && (3.18) \\
\forall i = 1, \ldots, n \quad \alpha_i &\geqslant 0 && (3.19) \\
\beta_i &\geqslant 0 && (3.20)
\end{aligned}$$

conditions KKT

$$\begin{aligned}
\alpha_i [y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 + \xi_i] &= 0 && (3.21) \\
\beta_i \xi_i = 0 = (\frac{C}{n} - \alpha_i)\xi_i \qquad &\text{(d'après (3.16))} && (3.22)
\end{aligned}$$

De manière intéressante, dans ce cas le problème dual s'écrit dans une forme analogue au cas des marges dures :

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

$$\text{soumis à} \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

$$\forall\, i = 1, \ldots, n \quad 0 \leqslant \alpha_i \leqslant \frac{C}{n}$$

On voit que la seule différence avec les SVM à marge dure porte sur la contrainte sur les α_i. En outre, (3.22) impose que, soit $\alpha_i = \frac{C}{n}$, soit $\xi_i = 0$. Les points mal classés sont donc identifiés par leur valeur α_i. Un autre avantage des marges souples $\mathbb{L}^1$ est que la solution du problème est plus souvent "creuse", *i.e.* une grande proportion des α_i est nulle.

La figure 3.4 résume la forme d'une solution avec les valeurs associées des α_i pour un problème simple dans $\mathbb{R}^2$.

Notons que les problèmes d'optimisation faisant intervenir des marges souples peuvent être aussi justifiés à l'aide d'une borne à la VC du même type que celle énoncée dans le théorème 3.6 et faisant intervenir explicitement les ξ_i.

Dans la pratique, les méthodes de calcul pour ce problème d'optimisation sont connues. Par exemple la méthode des points intérieurs optimise conjointement le problème et son dual, mais nécessite l'inversion de la matrice des produits scalaires, et ne peut donc être utilisée pour plus de quelques milliers d'exemples. Dans la pratique, pour les grands jeux de données, on utilise plutôt des algorithmes de type SMO (Sequential Minimal Optimisation), qui résolvent analytiquement le problème pour deux points et possèdent des vitesses de convergence satisfaisantes. Enfin, on peut aussi implémenter les méthodes de type "descente de gradient", qui, bien que coûteuses en temps de calcul, permettent l'apprentissage séquentiel du SVM (où de nouveaux exemples sont rajoutés dynamiquement).

3.3.3 Méthodes à noyau

Comme on a pu le constater, la solution du problème de minimisation présenté à la section précédente fait **uniquement** intervenir le produit scalaire entre les points d'entrainement dans $\mathbb{R}^m$. Ainsi, on pourrait envisager ne disposer que d'indices de similarité entre individus, pourvu que ces quantités correspondent bien à un produit scalaire. Un premier avantage est le surdimensionnement implicite de l'espace des attributs. Alors, l'ajustement d'un hyperplan de marge maximale, tout en garantissant par minimisation du risque structurel le meilleur compromis en terme de capacité, permettra d'obtenir des frontières de décision plus complexes.

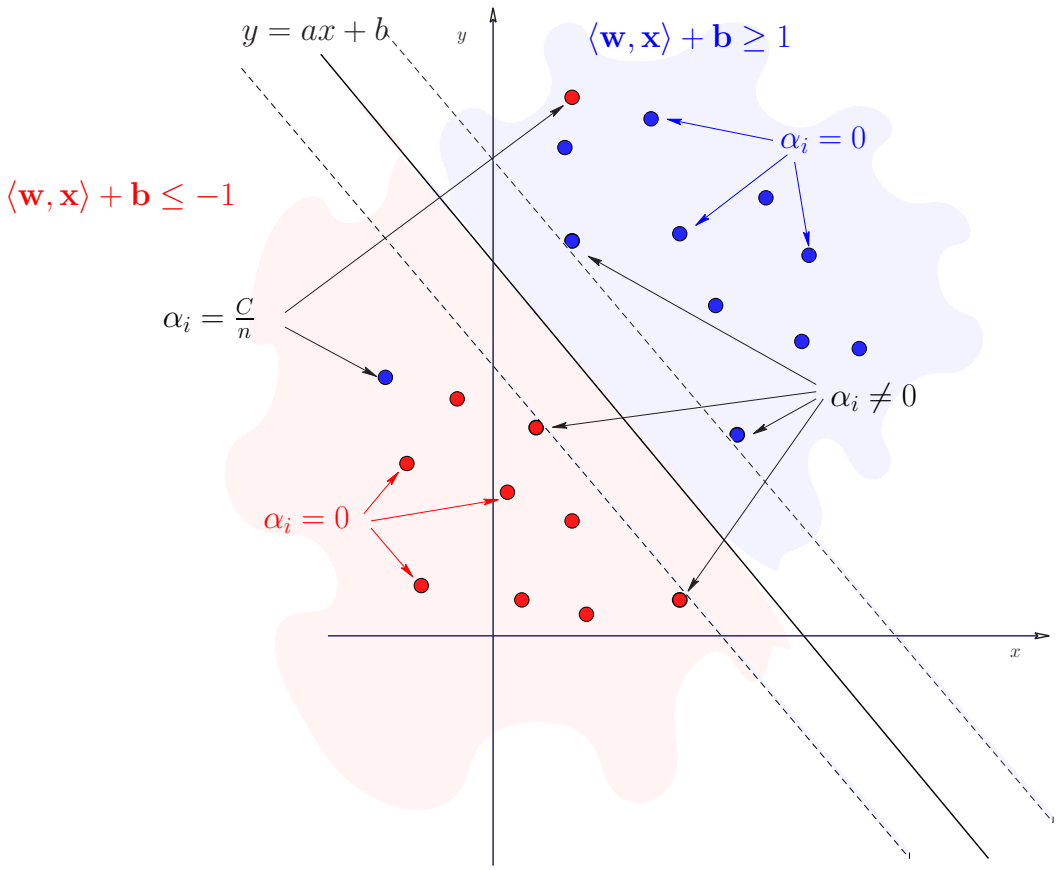


FIG. 3.4 – exemple d'un hyperplan séparateur et des valeurs α_i des points associés pour l'entraînement.

La matrice constituée de ces produits scalaires implicites est l'information nécessaire et suffisante pour résoudre le problème de minimisation. On appelle cette quantité la matrice de Gram. Donnons une définition plus générale qui nous permettra d'introduire le concept de noyaux et de voir à quelle condition cette matrice fait bien référence à un produit scalaire.

Définition 3.7 (Matrice de Gram). Soit $\mathcal{X}$ un ensemble non vide, et k une fonction de $\mathcal{X} \times \mathcal{X}$ dans $\mathbb{R}$. Pour $\mathbf{x}_1, \ldots, \mathbf{x}_m$ dans $\mathcal{X}$, la matrice K à m lignes et m colonnes définie par :

$$K(i, j) = k(\mathbf{x}_i, \mathbf{x}_j)$$

est appelée la matrice de Gram de k sur $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

Définition 3.8 (Noyau). Soit $\mathcal{X}$ un ensemble non vide, on appelle **noyau**[4] une fonction

[4] ou kernel en anglais. On préfèrera généralement utiliser ce terme pour éviter toute ambiguité avec les fonctions à noyau

de $\mathcal{X} \times \mathcal{X}$ dans $\mathbb{R}^+$ telle que

$$k(x_1, x_2) = k(x_2, x_1)\ \forall x_1, x_2 \in \mathcal{X} \qquad \text{(symétrique)} \qquad (3.23)$$
$$k(x_1, x_2) \geq 0\ \forall x_1, x_2 \in \mathcal{X} \qquad \text{(défini positif)} \qquad (3.24)$$

Une condition équivalente à (3.23,3.24) est que, quels que soient les points $x_1, \ldots, x_m$ dans $\mathcal{X}$, la matrice de Gram K associée au noyau k vérifie :

$$\sum_{i,j=1}^{m} c_i\, c_j\, K(i,j) \geq 0 \qquad \forall c_i, c_j \in \mathbb{R}$$

On voit facilement que tout produit scalaire dans $\mathbb{R}^m$ est un noyau. La section suivante va plus loin en montrant qu'à un noyau on peut associer un produit scalaire sur un espace de Hilbert. Ainsi, cela permettra de travailler sur des ensembles $\mathcal{X}$ quelconques et plus particulièrement les séquences de caractères (section 3.3.4).

3.3.3.1 Théorème de Mercer

Le théorème de Mercer permet de replonger explicitement n'importe quel noyau sur un espace mesurable dans un espace de Hilbert.

Théorème 3.9 (Mercer). *Soit $\mathcal{X}$ un espace mesurable de mesure μ finie et $k \in L_\infty(\mathcal{X}^2)$ une fonction symétrique à valeurs réelles telle que l'opérateur d'intégration T_k :*

$$T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$$
$$(T_k f)(x) := \int_{\mathcal{X}} k(x, x') f(x') d\mu(x')$$

est défini positif, c'est à dire que quel que soit $f \in L_2(\mathcal{X})$, on a :

$$\int_{\mathcal{X}^2} k(x, x') f(x) f(x') d\mu(x) d\mu(x') \geq 0 \qquad \textit{(noyau de Mercer)}$$

Alors, si $\left\{\psi_j\right\}_{j\geq 1}$ représente le jeu de fonctions propres orthonormales de T_k dans $\mathbf{L}_2(\mathcal{X})$ associées au valeurs propres $\lambda_j > 0$ et rangées en ordre non croissant, on a :

- *$(\lambda_j)_{j>0} \in \ell_1$ (i.e. $\sum_{j\geq 1} \left|\lambda_j\right| < \infty$)*
- *la série $S_{x,x'} = \sum_{j\geq 1} \lambda_j \psi_j(x) \psi_j(x')$ converge absolument et uniformément sur p.s. tout (x, x'). De plus on a $k(x, x') = S_{x,x'}$.*

Ce théorème permet donc de voir que pour tout noyau de Mercer, il existe une fonction Ψ de $\mathcal{X}$ dans un espace où k peut être vu comme un produit scalaire.

En d'autres termes, à tout noyau de Mercer, on peut associer un espace hilbertien muni d'un produit scalaire.

3.3.3.2 Exemples de noyaux sur $\mathbb{R}^d$

Le théorème de Mercer permet aussi de définir les kernels qu'on peut construire par opérations élémentaires :

Proposition 3.10. *Soient k_1 et k_2 des kernels sur $\mathcal{X} \times \mathcal{X}$, $a \in \mathbb{R}^+$, h une fonction réelle et positive sur $\mathcal{X}$, et Φ une fonction de $\mathcal{X}$ dans $\mathbb{R}^m$ à laquelle on peut associer un kernel k_3. Alors les fonctions suivantes sont des kernels :*

- $k_1(x, x') + k_2(x, x')$
- $ak_1(x, x')$
- $k_1(x, x')k_2(x, x')$
- $h(x)h(x')$
- $k_3(\Phi(x), \Phi(x'))$

Présentons quelques noyaux très fréquemment utilisés sur $\mathbb{R}^m$.

Noyau polynômial : ce noyau fait intervenir toutes les interactions terme à terme jusqu'au niveau l.

$$k_{(u,c,l)}(x, x') = (u\langle x, x'\rangle + c)^l$$

où les termes u et c permettent de corriger l'influence des $\binom{l}{k}$ quand l augmente. Par exemple pour $l = 2$, et $u, c = 0$, la fonction $\Phi(\mathbf{x})$ de $\mathbb{R}^d$ dans $\mathbb{R}^{d(d-1)/2}$ définie par $\Phi(\mathbf{x}) = \left(y_1, \ldots, y_{\frac{d(d-1)}{2}}\right)$ avec :

$$y_i = \begin{cases} x_{i/d}^2 & \text{si } d \text{ divise } i \\ \sqrt{2}x_{i\%d}x_{i/d} & \text{sinon} \end{cases}$$

est bien définie telle que $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$

Noyau gaussien : ou noyau RBF (pour Radial Basis Fonction) :

$$k_{(\sigma)}(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

Remarquons que $k(x, x) = 1$, et les points sont alors enserrés dans la boule unité. De plus ce noyau peut être en toute généralité combiné avec une norme sur $\mathcal{X}$.

Noyau sigmoide : de manière intéressante, ce noyau n'est pas défini positif, il donne cependant de bons résultats dans la pratique. En effet, la plupart des matrices de Gram construites à partir de données réelles définissent un sous espace où le noyau reste défini positif par combinaisons linéaires.

$$k(x, x') = \tanh(\kappa\langle x, x'\rangle + \vartheta)$$

3.3.3.3 Le kernel trick

Le problème d'optimisation présenté en section 3.3.2, se réécrit alors simplement pour un noyau k en adaptant le problème dual. Par exemple pour le cas des marges souples dans $\mathbf{L}^1$, cela donne :

$$\max_{\boldsymbol{\alpha}} W_k(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \, k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{soumis à} \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

$$\forall \, i = 1, \ldots, n \quad 0 \leqslant \alpha_i \leqslant \frac{C}{n}$$

et la fonctions de décision $f(\mathbf{x})$ est donnée par :

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} y_i \, \alpha_i k(x, x_i) + b \right)$$

où le décalage b est obtenu par l'utilisation des conditions KKT qui statuent que si $\alpha_j > 0$, on a :

$$\sum_{i=1}^{n} y_i \alpha_i k(x_j, x_i) + b = y_j$$

Cependant, on peut s'interroger sur la validité de cette démarche, par exemple dans le cas des noyaux gaussiens, où la capacité associée est de dimension infinie, et les bornes sur le risque telles que (3.1) ne sont plus suffisantes. Pour pouvoir montrer qu'alors cette approche est toujours justifiée, on doit avoir recours à des mesures de complexité sur la classe de classificateurs plus fines que la dimension VC, telles que les nombres de couverture, qui permettent alors d'obtenir le résultat voulu.

3.3.4 Noyaux pour séquences de caractères

On a pu voir que les méthodes à noyau permettaient de travailler en théorie avec n'importe quel ensemble, en conditionnant la géométrie sur les éléments à l'aide du kernel. Nous illustrons ici quelques noyaux qui ont été proposés sur les chaînes de caractères et qui ont ainsi été utilisés pour des problèmes de classification sur les séquences biologiques.

La section 3.3.4.1 présente deux noyaux pouvant être construits directement à partir des occurrences de motifs conjointes aux deux séquences. Les string-kernels comparent deux séquences sur la base des comptages des suites de lettres (non nécessairement contiguës) communes aux deux séquences. Ce cas permet d'écrire l'espace d'attributs

sur lequel le produit scalaire est calculé. On présentera aussi un noyau plus simple, le mismatch kernel, qui compte le nombre de motifs en commun entre deux séquences, en permettant au plus m substitutions. Ce dernier a donné des résultats encourageants en étant couplé à un classificateurs SVM sur des jeux de séquences protéiques.

Enfin, la section 3.3.4.2 présente une manière de définir des noyaux conditionnellement à un modèle probabiliste de génération des individus. Ces noyaux de Fisher ou des tangentes a posteriori créent ainsi un lien entre les méthodes d'analyse discriminante et les séparateurs à vaste marge. On montrera les applications aux modèles de Chaînes de Markov Cachées.

Dans la suite, on considérera un alphabet Σ de taille $|\Sigma|$. On notera Σ^ℓ l'ensemble des séquences de longueur ℓ et Σ^* l'ensemble de toutes les séquences $\Sigma^* = \cup_{i\geq 1}\Sigma^i$.

3.3.4.1 Mismatch et string kernels

string kernels Pour une séquence s, on notera la suite de ses éléments par $s(1)\dots s(n)$. La concaténation de s et d'une séquence t sera notée $s.t$. Regardons maintenant la forme des sous-séquences u d'une séquence donnée. Pour une suite d'index $\mathbf{i} = (i_1, \dots, i_{|u|})$ tels que $1 \leq i_1 < i_2 < \dots < i_{|u|} \leq n$ on définit u comme $s(\mathbf{i})$, c'est à dire la suite de lettres $s(i_1)s(i_2)\dots s(i_{|u|})$. On note alors $l(\mathbf{i}) = i_{|u|} - i_1 + 1$ la longueur de la sous séquence dans s, *i.e.* la dispersion globale des lettres dans s. Remarquons que si $\mathbf{i}$ est constituée d'index consécutifs, alors $l(\mathbf{i}) = |u|$.

Comme espace d'attribut construit à partir des séquences de longueur n, on prend $\mathbb{R}^{\Sigma^n}$, soit l'ensemble de toutes les fonctions de Σ^n dans $\mathbb{R}$, ou, de manière équivalente, l'espace où chaque séquence possible de longueur n occupe une dimension. On replonge alors une séquence s dans $\mathbb{R}^{\Sigma^n}$ avec l'application Φ_n définie pour chaque sous-séquence u de Σ^n par :

$$[\Phi_n(s)]_{u\in\Sigma^n} = \sum_{i \text{ tq } s(\mathbf{i})=u} \lambda^{l(\mathbf{i})}$$

où $0 < \lambda \leq 1$ est un paramètre influant sur la pondération imposée suivant la longueur des sous-séquences qui apparaissent dans s. la figure 3.5 montre par exemple la coordonnée GTC pour la séquence s =ACGTCTACGAGTCGT dans Σ^3.

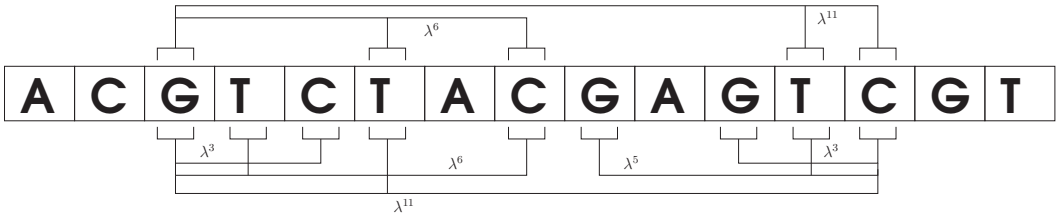


FIG. 3.5 – exemple pour $[\Phi_3(s)]_{\text{GTC}} = 2\lambda^3 + \lambda^5 + 2\lambda^6 + 2\lambda^{11}$.

Le kernel pour séquences de longueur n se déduit alors comme :

$$k_n(s,t) = \sum_{u \in \Sigma^n} [\Phi_n(s)]_u [\Phi_n(t)]_u = \sum_{u \in \Sigma^n} \sum_{(\mathbf{i},\mathbf{j}) \text{ tq } s(\mathbf{i})=s(\mathbf{j})=u} \lambda^{l(\mathbf{i})} \lambda^{l(\mathbf{j})} \qquad \forall s,t \in \Sigma^*$$

On peut bien sûr prendre en compte des séquences de longueurs différentes en combinant linéairement les k_n : $\sum_n c_n k_n$. De plus, comme la longueur des séquences influe sur la valeur du noyau, on renormalise en général les vecteurs par $\sqrt{k(s,s)k(t,t)}$.

Les quantités se calculent alors à l'aide de formules de récurrences sur $|s|$, $|t|$ et n, correspondant ainsi à une complexité en $o(n|s||t|)$ pour le calcul de $k_n(s,t)$ sur deux séquences s et t.

mismatch kernel Une approche classique concernant les travaux sur les chaînes de caractères, consiste à utiliser l'approche "sac de mots"[5], *i.e.* comparer deux séquences à partir des occurrences pour tous les mots d'une longueur fixée. Ceci est équivalent à l'utilisation du noyau "spectral"[6], qui, pour k fixé, replonge la séquence $\mathbf{S}$ dans $\mathbb{R}^{|\Sigma|^k}$ à l'aide de la fonction $\Phi_k(\mathbf{S}) = \left(N_a(\mathbf{S})\right)_{a \in \Sigma^k}$. Le kernel entre deux séquences $\mathbf{S}_1$ et $\mathbf{S}_2$ s'écrit donc :

$$K_k(\mathbf{S}_1, \mathbf{S}_2) = \sum_{a \in \Sigma^k} N_a(\mathbf{S}_1).\, N_a(\mathbf{S}_2)$$

Des extensions ont été proposées à ce noyau pour être adaptées plus particulièrement à la classification sur des protéines. Par exemple, le "mismatch" kernel présenté dans [LEC+04] compte les motifs communs entre deux séquences en permettant jusqu'à m substitutions entre eux. Plus précisement, en notant $N_a(\mathbf{S}, m)$ le nombre d'occurrences de a dans $\mathbf{S}$ en permettant au plus m substitutions. Le mismatch kernel $K_{k,m}$ est défini sur $\mathbf{S}_1$ et $\mathbf{S}_2$ par :

$$K_{(k,m)}(\mathbf{S}_1, \mathbf{S}_2) = \sum_{a \in \Sigma^k} N_a(\mathbf{S}_1, m) N_a(\mathbf{S}_2, m)$$

A nouveau dans ce cas, les données seront en général renormalisées par $\sqrt{k(x,x)k(y,y)}$.

Ces noyaux génèrent des vecteurs d'attributs de grandes dimensions, mais particulièrement creux. L'implémentation fait donc appel à des objet algorithmiques évolués tels que les arbres de suffixe afin de permettre des calculs linéaires sur la longueur des séquences. On pourra regarder [Gus97] pour plus de détails sur ces objets.

Des noyaux, appelé souvent noyaux à motifs, ont été aussi proposés pour l'annotation fonctionnelle sur les protéines. On utilise alors la connaissance biologique disponible en ne comptant que les motifs protéiques annotés dans les banques de données lors de la comparaison de deux séquences.

[5] bag of words [6] traduction libre de "spectrum kernel"

3.3.4.2 Noyaux probabilistes (Fisher et TOP kernels)

Les noyaux que nous venons de présenter construisent explicitement un espace d'attributs fondés sur les comptages des mots dans la séquence. Comme nous l'avons vu précédemment (section 3.2), cette extraction d'attributs correspond dans une certaine mesure à l'estimation de paramètres pour un modèle probabiliste imposé sur les séquences. De plus, dans le chapitre 2, nous avons pu voir que les scores et la matrice de Fisher induisaient une géométrie dans l'espace de la log-vraisemblance. Les noyaux de Fisher et TOP proposent ainsi de construire un kernel **conditionnellement** à un modèle présidant à la génération des individus.

Noyau de Fisher Le noyau de Fisher présenté dans [JDH00] et [JH99] définit une mesure de similarité dans l'espace des paramètres du modèle, représentés par leur score de Fisher, pour la métrique définie par l'information de Fisher. Plus formellement, il est défini comme suit :

Définition 3.11. Soit x et x' deux réalisations sous un modèle paramétré par $\boldsymbol{\theta}$, le noyau de Fisher $\mathrm{K}_{\mathcal{F}}$, à valeurs dans $\mathbb{R}^+$ est défini comme :

$$\mathrm{K}_{\mathcal{F}}(x, x') = \mathcal{S}_{\mathcal{F}}(x, \boldsymbol{\theta})^{\top} . \mathcal{I}^{-1}(\boldsymbol{\theta}) . \mathcal{S}_{\mathcal{F}}(x', \boldsymbol{\theta})$$

où $\mathcal{S}_{\mathcal{F}}$ est le score de Fisher, *i.e* :

$$\mathcal{S}_{\mathcal{F}}(x, \boldsymbol{\theta}) = (\partial_{\theta_1} \mathcal{L}(x, \boldsymbol{\theta}), \ldots, \partial_{\theta_d} \mathcal{L}(x, \boldsymbol{\theta}))^{\top} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(x, \boldsymbol{\theta})$$

et $\mathcal{I}(\boldsymbol{\theta})$ la matrice d'information de Fisher attendue. Dans la pratique, de par la difficulté de ce calcul, cette matrice est souvent omise, et remplacée par la matrice identité de dimension d, ou la matrice des variances par dimension : $\left(\delta_{ij} . \sigma_i^2\right)_{i,j=1}^{d}$

Dans le cas d'une CMC, le calcul des dérivées de la log-vraisemblance d'une séquence se calcule à partir des variables forward et backward obtenues par l'algorithme de Baum-Welsch (voir 2.1.2.2 et 2.2.2.4).

Noyau TOP Ce noyau a été proposé dans la continuité du noyau de Fisher, mais fondé sur une approche plus "pragmatique" du problème. Revenons au problème de classification dans le cas de l'analyse discriminante pour présenter le noyau qui peut en être déduit. La classe d'un individu x est déterminée par :

$$\arg \max_{i \in \{-1;+1\}} \left(\mathbf{P}(Y = i) \mathbf{P}(X = x | Y = i) \right)$$

La fonction de décision $f(x)$ correspondant à la loi de Bayes s'écrit pour le vrai jeu de paramètres $\boldsymbol{\theta}^*$:

$$f(x) = \mathrm{sgn}\left\{ \mathbf{P}(y = +1 | x, \theta^*) - \mathbf{P}(y = -1 | x, \theta^*) \right\} \tag{3.25}$$

$$= \mathrm{sgn}\left\{ \alpha \mathbf{P}(x | y = +1, \theta^*) - (1 - \alpha) \mathbf{P}(x | y = -1, \theta^*) \right\} \tag{3.26}$$

Le facteur en $\mathbf{P}(x|\theta^*)^{-1}$ se simplifiant, et α correspondant à l'a priori sur la classe +1. Si on considère le logarithme du rapport des probabilités a posteriori :

$$v(x, \boldsymbol{\theta}^*) = \log \frac{\mathbf{P}(y = +1|x, \boldsymbol{\theta})}{\mathbf{P}(y = -1|x, \boldsymbol{\theta})} \tag{3.27}$$

$$= \log \left(\frac{\alpha \mathbf{P}(x|y = +1, \boldsymbol{\theta}^*)}{(1 - \alpha)\mathbf{P}(x|y = -1, \boldsymbol{\theta}^*)} \right) \tag{3.28}$$

$$= \log \mathbf{P}(x|y = +1, \boldsymbol{\theta}^*) - \log \mathbf{P}(x|y = -1, \boldsymbol{\theta}^*) + \log \left(\frac{\alpha}{1 - \alpha} \right) \tag{3.29}$$

La classe de la séquence est maintenant déterminée par le signe de $v(x, \boldsymbol{\theta}^*)$. Dans la pratique $\boldsymbol{\theta}^*$ n'est pas accessible, mais on peut approximer $v(x, \boldsymbol{\theta}^*)$ par un développement limité autour de $\hat{\theta}$:

$$v(x, \boldsymbol{\theta}^*) = v(x, \hat{\boldsymbol{\theta}}) + \langle \nabla_{\boldsymbol{\theta}} v(x, \hat{\boldsymbol{\theta}}) | (\boldsymbol{\theta}^* - \boldsymbol{\theta}) \rangle + o(\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|) \tag{3.30}$$

$$\approx v(x, \hat{\boldsymbol{\theta}}) + \langle \nabla_{\boldsymbol{\theta}} v(x, \hat{\boldsymbol{\theta}}) | (\boldsymbol{\theta}^* - \boldsymbol{\theta}) \rangle \tag{3.31}$$

$$=: \langle \mathcal{S}_{\mathcal{T}}(\hat{\boldsymbol{\theta}}, x) | \mathbf{w} \rangle \tag{3.32}$$

avec :

$$\mathcal{S}_{\mathcal{T}}(\hat{\boldsymbol{\theta}}, x) = (v(x, \hat{\boldsymbol{\theta}}), \partial_{\theta_1} v(x, \hat{\boldsymbol{\theta}}), \ldots, \partial_{\theta_d} v(x, \hat{\boldsymbol{\theta}}))^\top \tag{3.33}$$

$$\mathbf{w} = (1, \theta_1^* - \hat{\theta}_1, \ldots, \theta_d^* - \hat{\theta}_d)^\top \tag{3.34}$$

Ainsi, le problème de classification se résume à l'utilisation d'un classificateur linéaire pour déterminer $\mathbf{w}$. Remarquons que dans le même temps, le terme de biais $\log(\frac{\alpha}{1-\alpha})$ est aussi déterminé par le classificateur ne nécessite plus une estimation empirique.

Ayant réalisé explicitement la projection dans un espace des attributs, la définition du noyau TOP s'écrit directement :

Définition 3.12. Le produit scalaire, noté $\mathrm{K}_{\mathcal{T}}$ et défini pour deux séquences x et x' par :

$$\mathrm{K}_{\mathcal{T}}(x, x') = \left\langle \mathcal{S}_{\mathcal{T}}(\hat{\theta}, x), \mathcal{S}_{\mathcal{T}}(\hat{\theta}, x') \right\rangle$$

est appelé le noyau TOP, pour Tangent Of the Posterior logg-odds (la tangente des log vraisemblances a posteriori), car c'est la composante expliquant la majeure partie du noyau.

Dans le cas où les individus sont réellement générés par les données, on peut démontrer (voir [TKR+02]) que le risque du noyaux TOP converge plus vite vers le risque optimal que celui obtenu par analyse discriminante, en assumant que le séparateur linéaire optimal est choisi.

3.3.5 Méthodes de détermination automatique des paramètres d'un noyau

Malgré les grands avantages proposés par la méthode SVM, tels que l'existence d'une unique solution optimale et la stratégie de minimisation du risque qui ne fait pas d'hypothèse sur la forme des données à minimiser ; un certain nombre de paramètres sont à fixer :

- le paramètre C de coût pour les points qui sont mal classés.
- les paramètres liés au noyau. Par exemple dans le cas d'un noyau polynomial, on doit déterminer l, c et u ; ou σ pour les noyaux RBF.

Dans la suite on supposera donc qu'on se donne un noyau $k_{\boldsymbol{\theta}}$ dépendant d'un jeu de paramètres $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_\ell)$.

3.3.5.1 Détermination par grille

La méthode la plus courante consiste alors à se fixer une grille de valeurs possibles : pour C dans un intervalle fixé $[C_{\min}, C_{\max}]$ avec des incréments ρ_C ; pour chacun des θ_i dans un intervalle $\left[\theta_i^{\min}, \theta_i^{\max}\right]$ avec des incréments ρ_{θ_i}. Alors, pour chaque jeu de paramètres $\{C, \boldsymbol{\theta}\}$ on estime le risque du SVM associé par validation croisée : $R^{\mathrm{CV}}_{(C,\boldsymbol{\theta})}$. On sélectionne ensuite le jeu de paramètres $\left\{\bar{C}, \bar{\boldsymbol{\alpha}}\right\}$ ayant réalisé le risque minimum. Remarquons qu'on peut aussi prendre un autre indice pour le classificateur lors de la sélection des paramètres. Par exemple, si les classes présentent de grandes différences d'effectifs, on pourra sélectionner le classificateur présentant le meilleur taux de bien classés sur une des classes.

Remarquons que cette méthode nécessite d'entraîner un nombre de SVM croissant exponentiellement avec le nombre de paramètres pris en compte. En effet si on permet k valeurs par paramètre, on doit alors tester k^ℓ paramètres, et le coût de calcul devient vite impraticable.

3.3.5.2 Minimisation du risque LOO

Comme alternative [CVBM02] proposent une procédure itérative permettant de travailler sur une borne de l'erreur "Leave One Out". Comme on l'a vu au début de ce chapitre, malgré le fait que sa variance ne soit pas caractérisée, le risque LOO est une borne supérieure non biaisé sur le risque d'un classificateur (par exemple en section 3.1). En outre, on a :

$$R^{\mathrm{LOO}}(w) \leq R^2 \left\|w^2\right\|$$

Les résultats présentés par la suite s'utilisent dans le cas des SVM à marge dure. Cependant, on voit rapidement qu'en redéfinissant la matrice de Gram des exemples $K_{\boldsymbol{\theta}}$ par $\tilde{K}_{\boldsymbol{\theta},C} = K_{\boldsymbol{\theta}} + \frac{1}{C} I$, résoudre le problème à marges dures pour $\tilde{K}$ revient à résoudre le SVM à marges souples dans $\mathbf{L}^2$ pour K.

Ainsi, pour résumer le problème, calculer le paramètre à $\boldsymbol{\theta}$ fixé revient à maximiser la fonctionnelle du problème duale :

$$\begin{aligned} \arg\max_{\boldsymbol{\alpha}} W_{\boldsymbol{\theta}}(\boldsymbol{\alpha}) &= \sum_{i=1}^{L} \alpha_i - \sum \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k_{\boldsymbol{\theta},C}(x_i, x_j) \\ \text{soumis à} \quad & \sum_{i=1}^{L} \alpha_i y_i = 0 \\ \text{et} \quad & \alpha_i \geq 0 \quad \forall i = 1, \ldots, L \end{aligned}$$

La méthode proposée dans [CVBM02] propose d'alterner des étapes de résolution du SVM à θ fixé avec des descentes de gradient sur $\boldsymbol{\theta}$ pour la borne $R^2 \left\|w^2\right\|$. Ainsi, pour un critère $T(\boldsymbol{\alpha}, \boldsymbol{\theta})$ obtenu à $\boldsymbol{\alpha}_0$ fixé, on veut calculer la dérivée T par rapport à $\boldsymbol{\theta}$.

Dans notre cas, on prend $T = R^2 \|w\|^2$ et donc on veut calculer :

$$\frac{\partial T}{\partial \theta_k} = \frac{\partial R^2}{\partial \theta_k} \|w\|^2 + R^2 \frac{\partial \|w\|^2}{\partial \theta_k}$$

Un lemme permet de calculer ces dérivées en utilisant le fait que la borne sur l'erreur s'exprime en fonction de la solution du problème SVM à chaque étape.

Lemme 3.13. *Soit $v_\theta \in \mathbb{R}^{n+1}$ et P_θ une matrice carrée de dimension n^2 dépendant continûment d'un paramètre θ. Soit la fonction :*

$$\begin{aligned} L(\theta) &= \max_{\mathbf{x} \in F} \langle \mathbf{x}, \mathbf{v}_{\boldsymbol{\theta}} \rangle - \frac{1}{2} \mathbf{x}^\top P_\theta \, \mathbf{x} \qquad (3.35) \\ \textit{avec} \quad & F = \{\mathbf{x} / \langle \mathbf{b}, \mathbf{x} \rangle = c; x \geq 0\} \end{aligned}$$

Soit $\bar{\mathbf{x}}$ le vecteur où le maximum sur F en $L(\theta)$ est atteint. Si $\bar{\mathbf{x}}$ est unique, on a alors :

$$\frac{\partial L(\theta)}{\partial \theta} = \left\langle \bar{\mathbf{x}}, \frac{\partial \mathbf{v}_\theta}{\partial \theta} \right\rangle - \frac{1}{2} \bar{\mathbf{x}}^\top \frac{\partial P_\theta}{\partial \theta} \bar{\mathbf{x}}$$

En d'autres termes, L est différentiable par rapport à θ sans tenir compte de la dépendance de $\bar{\mathbf{x}}$ en θ. Le résultat reste valable si une ou les deux contraintes définissant F sont relâchées.

Pour la démonstration de ce lemme et des résultats équivalents, on pourra regarder par exemple dans [BS00]. L'intérêt ici est que R^2 et $\left\|w^2\right\|$ sont tous deux solutions d'un problème d'optimisation du type de (3.35), et leurs dérivées sont donc calculables :

$$\frac{\partial \|w\|^2}{\partial \theta_k} = -\sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i^0 \alpha_j^0 y_i y_j \frac{\partial k_\theta(x_i, x_j)}{\partial \theta_k} \qquad (\text{car } \|w\|^2 = 2W(\boldsymbol{\alpha}^0)) \qquad (3.36)$$

$$\frac{\partial R^2}{\partial \theta_k} = \sum_{i=1}^{L} \beta_i^0 \frac{\partial k_\theta(x_i, x_i)}{\partial \theta_k} - \sum_{i=1}^{L} \sum_{j=1}^{L} \beta_i^0 \beta_j^0 \frac{\partial k_\theta(x_i, x_j)}{\partial \theta_k} \qquad (\text{avec } R^2 = S(\boldsymbol{\beta}^0)) \qquad (3.37)$$

Le premier intérêt de cette méthode est de permettre a priori un moins grand nombre d'entraînement de SVM que la méthode par grille pour la détermination des paramètres optimaux. En outre, cette approche permet aussi en théorie de permettre de la sélection de variables en rajoutant comme paramètres du noyau des facteurs d'échelle pour chacune des dimensions de la variable. Par exemple, pour des exemples $\mathbf{x}$ et $\mathbf{y}$ dans $\mathbb{R}^d$, on pourrait redéfinir un noyau gaussien avec un vecteur de poids par dimension $\boldsymbol{\sigma}$ comme :

$$k_{\boldsymbol{\sigma}}(x, y) = \exp\left(\sum_{i=1}^{d} \frac{|x_i - y_i|}{2\sigma_i}\right)$$

et procéder à la descente de gradient sur $\boldsymbol{\sigma}$.

D'autres méthodes de détermination automatique des hyperparamètres du noyau pour les SVM ont été développées en étendant les résultats de Chapelle [DSA03], ou par exemple l'introduction d'autres méthodes comme dans [LCB+04], [CSTEK02] ou dernièrement [OSW05]. Par la suite nous nous sommes contentés d'utiliser la méthode présentée ci dessus quand c'était possible.

Bibliographie

[BS00] J.F. Bonnans and A. Shapiro. Perturbation analysis of optimisation problems, 2000.

[CSTEK02] Nello Cristianini, John Shawe-Taylor, André Elisseeff, and J. Kandola. *Advances in Neural Information Processing Systems 14*, chapter On Kernel-Target Alignment, pages 367–373. MIT Press, 2002.

[CVBM02] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1) :131–159, 2002.

[DSA03] K. Duan, Keerthi S.S, and Poo A.N. Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51 :41–59, 2003.

[Gus97] Dan Gusfield. *Algorithms on Strings, Trees and Sequences : Computer Science and Computational Biology.* Cambridge University Press, 1997.

[JDH00] T.S. Jaakkola, M. Diakhans, and D. Haussler. A discriminative framework for detecting protein remote homologies. *Journal of Computational Biology*, 7 :95–114, 2000.

[JH99] T.S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In S.A. Solla M.S. Kearns and editors D.A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1999.

[LCB+04] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5 :27–72, 2004.

[LEC+04] Christina S. Leslie, Eleazar Eskin, Adiel Cohen, Jason Weston, and William Stafford Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4) :467–476, 2004.

[NN92] H Nakashima and K Nishikawa. The amino acid composition is different between the cytoplasmic and extracellular sides in membrane proteins. *FEBS Letter*, 303 :141–146, 1992.

[OSW05] C.S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6 :1043–1071, 2005.

[SS01] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[TKR+02] K Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.R. Müller. A new discriminative kernel from probabilistic models. In G. Dietterich, S. Becker, and Z. editors Ghahramani, editors, *Advances in Neural Information Processing Systems*. MIT Press, 2002.

Troisième partie

Localisation subcellulaire des protéines

Chapitre 4

Jeux de données

Avant d'aborder les travaux réalisés, je présenterai dans ce chapitre les simplifications biologiques qui sont supposées pour l'élaboration du jeu de données.

Revenons aux connaissances biologiques énoncées dans le chapitre 1. Si les compartiments biologiques permettent la réalisation de tâches variées à l'intérieur de la cellule par cloisonnement spatial, les dynamiques cellulaires contredisent la vision simpliste d'une protéine résidant statiquement dans un compartiment. Par exemple, les facteurs de transcription, qui résident dans le cytoplasme, sont importés dans le noyau à la suite de stimuli extérieurs. De même, les protéines de régulation de l'ARN cyclent fréquemment entre le noyau et le cytosol.

Bien que la multicompartimentalité potentielle d'une protéine apporte une information substantielle pour son annotation, le manque actuel d'informations disponibles dans les bases de données rend ce type de prédictions difficile, certains auteurs s'étant déjà penchés sur cette question [STH04].

En outre, l'augmentation de la quantité de données concernant les séquences de transcrits montre que les phénomènes d'épissage alternatif constituent un facteur de diversité plus fréquent que ce que l'on pouvait penser auparavant. En effet, pour ne prendre qu'un exemple, lors de la création de la "H-Invitationnal Database" [IIS+04], qui rassemblait à sa création les informations concernant 41,118 transcrits pleine longueurs validant 21,037 gènes, la recherche sur 7,874 d'entre eux montrait que 40% correspondaient à un épissage alternatif. Sur ces 3,181 gènes, une moyenne de 2.7 transcrits différents par région codante était observée. Les phénomènes d'épissage alternatif pourraient donc de par la modularité des signaux d'adressage, influer sur l'adressage d'une protéine. Nakao et Al. ont étudié les variations en acides aminés au sein de ces formes alternatives sur plus de 2600 gènes humains extraits de swissprot, la H-invitationnal Database et la littérature [NBM+05]. Dans cette étude, à notre connaissance la seule à ce jour sur ce sujet, les auteurs ont examiné l'influence que pouvait avoir l'épissage alternatif des extrémités N et C terminales d'un transcrit sur la présence de signaux d'adressage. Par prédictions à l'aide de TargetP, et PTS1 (voir 5.1 pour plus de détails),

les auteurs ont examiné l'influence qu'on pouvait attendre pour les signaux peptides et les signaux d'adressage à la mitochondrie ou au péroxisome. Les résultats sont résumés dans la table 4.1.

Signal d'adressage	proportion de gènes où le signal change, (# positifs) (# négatifs)
signal d'adressage au péroxisome de type 1	56 % (61) (48)
Signal d'adressage à la mitochondrie	55 % (63) (51)
Signal peptide	22 % (124) (432)

TAB. 4.1 – Variations prédites sur la présence ou absence de signaux d'adressage. Remarquons que le signal de type 1 pour le péroxisome est recherché sur l'extrémité C-terminale de la protéine à la différence des signaux peptides et d'adressage à la mitochondrie qui se trouvent sur l'extrémité N-terminale.

Comme on peut le constater, à l'exception du signal peptide, la moitié des sites sont attendus pour présenter une variation dans la présence ou l'absence du signal d'adressage. Ajoutons que l'étude du nombre d'hélices transmembranaires à l'aide du programme THMM prédit que 27 % (=183/667) changent de nombre d'hélices transmembranaires pour les isoformes alternatives. Ces résultats sont en accord avec une étude portant sur 31 protéines mitochondriales connues pour être localisées dans différents compartiments [MAPM04]. A l'exception d'un exemple, la diversité compartimentale des protéines étudiées était due à des formes différentes de transcrits.

Hélas, la prise en compte des formes alternatives n'est pas encore d'actualité lors de la mise au point d'une stratégie de prédiction de la localisation cellulaire.

En raison des stratégies de classification que nous avons adoptées, les trois hypothèses suivantes seront donc faites pour la création du jeu de données :

Postulat 1 : Le compartiment d'une protéine fait référence au compartiment où celle ci est finalement adressée et fonctionnelle (un facteur de transcription sera annoté comme appartenant au noyau).

Postulat 2 : Sauf indication contraire, les protéines sont supposées appartenir à un seul compartiment. Les protéines annotées comme présentes dans plusieurs compartiments seront retirées à la création des jeux de données.

Postulat 3 : Chaque région codante est associée à une unique protéine dans le jeu de données. Si une protéine possède des isoformes alternatives, une seule sera choisie.

Dans la pratique, les auteurs assument implicitement que les isoformes alternatives sont retirées lors de la vérification sur les protéines très similaires.

Cette phase de regroupement puis d'élimination des protéines possédant un fort pourcentage d'identité de séquence est essentielle pour la création d'un jeu de données

non biaisé. En effet, une grande partie des informations utilisées sont issues de la séquence primaire pour prédire la localisation. En outre, deux protéines homologues (en excluant le cas de l'épissage alternatif considéré précédemment) tendent à être adressées dans le même compartiment cellulaire (voir 5.3 pour plus de détails). On peut donc penser que les jeux de données contenant encore une forte proportion de séquences homologues et utilisant directement des propriétés extraites de la séquence biologique, transgresseraient le principe d'indépendance des individus du jeu de données. Ce point n'est pas traité avec le même sérieux suivant les publications. Comme on le verra dans la section suivante, le soin apporté à cette étape peut influer fortement sur le jeu de données construit et donc sur la qualité des résultats présentés.

Enfin, il est nécessaire de définir une ontologie sur les compartiments qui seront pris en compte, afin de pouvoir associer à chaque annotation une classe précise. Les douze compartiments retenus tous eucaryotes confondus sont : membrane plasmique, noyau, extracellulaire, chloroplast, mitochondrie, cytoplasme, lysosome, cytosquelette, Reticulum Endoplasmique (ER), péroxisome, vacuole et appareil de Golgi. D'autre compartiments mériterait d'être pris en compte, comme par exemple l'ERGIC (ER to Golgi Intermediate Compartment), mais les données disponibles sont actuellement insuffisantes. Remarquons aussi que le cytosquelette n'est pas à proprement parler un compartiment.

Jeux de données	Nombre de séquences	Nombre de compartiments	Qualité d'annotation	% similarité max. (programme)	séquence nucléique ?
Park& Kanehisa	7579	12	swissprot (0)	80% (DIALIGN)	non
homme	2240	9	Hera (++)	100%	oui (cDNA)
levure	1491	9	SGD (+)	40% (FASTA)	oui (SGD)
Souris	1382	5	swissprot (0)	100%	oui (cDNA)

TAB. 4.2 – résumé des caractéristiques des différents jeux de données construits pour les expérimentations. Le jeu noté Park & Kanehisa est tiré de [PK03] et sera présenté en section 4.1.2. Les notations entre parenthèses dans la colonne qualité d'annotation donnent une indication sur le niveau de vérification des informations, de la simple annotation swissprot (0) à celui où toutes les données ont été vérifiées par plusieurs sources concordantes dans la littérature (++).

Le tableau 4.2 ci-dessus résume quelques informations relatives aux caractéristiques des jeux de données que nous avons utilisées lors des expérimentations. Les détails de mise au point de ces différents jeux seront présentés dans les sections suivantes. Remarquons qu'on a désiré travailler avec des jeux présentant des particularités différentes. En effet, malgré le caractère généraliste et contestable de cette approche, le premier jeu de données correspond aux protéines annotées de la base de données swissprot, de la même manière qu'une partie des méthodes proposées pour la prédiction de la localisation cellulaire (chapitre suivant). Ensuite, deux jeux de données spécifiques à une

espèce ont été pris en compte, les deux différences majeures étant l'attention qui a été accordée à la vérification des annotations de localisation, et l'élimination effectuée sur les protéînes trop similaires. Enfin, dans le seul but de proposer une vérification pour les résultats obtenus sur l'utilisation des fréquences de codons pour la classification, des jeux ont été construit sur la Souris et l'Homme, utilisant des alignements pleine longueur avec les banques de cDNA pour la détermination de la séquence nucléique.

4.1 Jeux déduits de Swissprot

4.1.1 méthode de mise au point

Cette courte section présente les points commun entre les plupart des jeux de données qui seront présentés en sections 5.2 et 5.3. En effet, la plupart sont construits en utilisant l'information mise à disposition dans Swissprot. Nous verrons dans la section suivante les détails pour la construction du jeu de données que nous avons utilisé, et quelques unes de ses particularités par comparaison avec un autre jeu.

Ces jeux de données sont construits en suivant les étapes suivantes :

1. Extraction des protéines possédant l'annotation "SUBCELLULAR LOCALIZATION" dans le champs de commentaire (CC).
2. Elimination des protéines où ledit champs est annoté comme "POSSIBLE", "PROBABLE", "SPECIFIC PERIODS" ou "BY SIMILARITY".
3. Elimination des protéines possédant plusieurs annotations de localisation, et de certains cas particuliers, comme par exemple quand la localisation de la protéine n'est annotée que pour certains tissus.
4. Attribution d'une classe pour chaque protéine, en accord avec l'ontologie choisie.
5. (optionnel) Nettoyage des protéines présentant des séquences avec un fort pourcentage de similarité de séquence à l'aide d'un logiciel d'alignement tel que BLAST, FASTA ou DIALIGN (exception faite des méthodes utilisant l'homologie).

4.1.2 Jeu de Park & Kanehisa (Swissprot v. 41.0)

Ce jeu de données est déduit de la base de données Swissprot version numéro 41. Il a été mis au point pour les résultats présentés dans l'article de [PK03] et est composé de 7579 séquences réparties sur tous les règnes. Ce jeu de données a été choisi car c'est l'un des seul proposant de travailler avec un nombre de séquences par compartiments suffisant.

Compartiment	SVM Keun-Joon P.		fuzzy k-NN	
	sensibilité	effectifs	sensibilité	effectifs
Plasma membrane	92.2	1677	–	–
Nuclear	89.6	1932	81.9	2152
Extracellular	78.0	862	93.7	2135
Chloroplast	72.3	671	84.7	645
Cytoplasmic	72.2	1245	70.2	1251
Lysosomal	61.8	93	67.5	83
Cytoskeleton	58.5	41	40.0	10
Mitochondrial	57.4	727	59.0	692
ER	46.5	114	57.3	82
Peroxisomal	25.2	125	56.8	81
Vacuolar	25.0	54	34.1	41
Golgi apparatus	14.6	48	16.1	31
Sensibilité globale	78.2		80.1	

TAB. 4.3 – Comparatifs des résultats obtenus à l'aide des deux méthodes utilisant des jeux de données obtenus sur Swissprot version 41.0, mais avec une méthode de nettoyage des protéines redondantes différents. Les sensibilités des méthodes sont données pour information.

Le jeu a été réduit de manière à assurer une similarité de séquence d'au plus 80% en alignement pleine longueur entre deux protéines quelconques du jeu de données. Les alignements ont été faits à l'aide d'une modification du programme DIALIGN.

Néanmoins, malgré ce "nettoyage", un grand nombre d'homologues semble subsister. En effet, les identifiants swissprot sont constitués de deux parties, séparées par un underscore '_'. La première partie identifie la protéine et la seconde spécifie l'information sur l'espèce. En approximant l'homologie entre deux protéines comme l'égalité sur la première partie de leur identifiant, près de la moitié des séquences (3766) possèdent au moins un homologue dans le jeux de données. En outre, 20 % (1373) font partie d'un groupe d'au moins 5 homologues. Dans le cas de ce jeu de données, la relation d'homologie conserve fortement l'annotation de localisation cellulaire pour la protéine. En effet, seulement 4,6% des protéines (173) seraient mal classées en utilisant l'information de l'appartenance au même groupe d'homologues.

Insistons aussi sur l'influence que peut avoir la valeur du seuil fixé lors de la réduction des homologues par l'examen d'un jeu de données déduit de la même version de Swissprot [HL04]. Dans ce second cas, les auteurs se sont basés sur la proportion d'acides aminés identiques après un alignement par BLAST, le seuil étant fixé à 80%. Les effectifs et résultats publiés sont listés dans la table 4.3. et montrent des effectifs

très variables suivant la méthode de regroupement utilisée. Sous l'hypothèse que les effectifs observés dans les compartiments sont générés par une même loi multinomiale (après retrait des protéines de la membrane plasmique), un test par rapport de vraisemblance est rejeté avec une significativité de 10^{-9}.

4.2 Jeux limités à une espèce

4.2.1 Jeu Homo-Sapiens (Hera)

Ce jeu est restreint à l'espèce humaine, et a été utilisé pour prédire la localisation d'une protéine à partir des motifs Interpro détectés dans la séquences (voir [STH04] et la section 5.2.2). Il a été mis au point à partir de la base de données Hera [SLHT04] originellement spécialisée sur les protéines adressées dans le réticulum endoplasmique, puis étendue par la suite aux autres compartiments. Il est présenté comme composé de 2216 protéines, réparties sur 9 compartiments, dont 18 protéines annotées comme multicompartimentales. Notons que la localisation de chacune des protéines présentes dans ce jeu de données a été confirmée manuellement par vérification à partir de plusieurs informations convergentes (expériences, littérature, bases de données).

Pour nos tests, M. Scott nous a fourni son jeu de données du 1er juillet 2004 composé de 2394 protéines réparties sur 14 compartiments (certaines protéines étaient disponibles avec des annotations du type ERGIC, périphérie du RE, vésicules de transport ou Endosome). Après retrait des protéines annotées comme multicompartimentales et des compartiments de trop faible effectif, notre jeu de données est composé de 2240 protéines réparties sur 9 compartiments (les effectifs de chaque compartiment sont listés dans la table 4.4).

4.2.2 Jeu *S. Cerevisiae*

Ce jeu est déduit des annotations expérimentales présentes sur la *Saccharomyces genome database*[1] en utilisant les annotations Gene Ontology mises à disposition. A titre de vérification, nos annotations ont été comparées à celle compilées par M. Scott et sont similaires sur plus de 95% des protéines. Comme la levure est connue pour posséder un grand nombre de paralogues, et pour voir l'influence que peuvent avoir les protéines similaires sur la qualité de la classification, les protéines présentant plus de 40% de similarité en pleine longueur ont été regroupées en clusters (voir section 7.1.1). Une protéine est ensuite prise au hasard parmi chaque cluster. Tous les alignements deux à deux ont été réalisés à l'aide du logiciel FASTA, avec comme matrice de substitution BLOSUM62. La table 4.4 résume les effectifs de chacun des compartiments avant et après l'élimination des homologues.

[1] `http ://www.yeastgenome.org`

compartiment	Homme	levure SGD	levure sans redondance
Cytoplasme	348	248	139
Cytosquelette	-	150	87
Réticulum Endoplasmique	326	175	133
Golgi	90	82	58
Lysosome	91	-	-
Mitochondrie	269	389	289
Noyau	581	933	651
Péroxisome	36	39	25
Vacuole	-	75	51
Membrane plasmique	205	125	58
extracellulaire	294	-	-
total	2240	2216	1491

TAB. 4.4 – résumé des effectifs pour les jeux de données sur l'homme et sur la levure.

Remarquons que deux stratégies à grande échelle ont été menées durant les années précédentes sur la levure [HFG+03, KAH+02]. Dans [HFG+03], les auteurs affirment même pouvoir annoter la localisation sur 22 compartiments pour 75% du protéome de la levure. A l'exception de résultats préliminaires sur l'usage des codons chez la levure, qui seront présentés dans la section 7.1.1, nous ne prendrons pas en compte ces données pour les résultats présentés au chapitre 7.2.

4.2.3 Jeux de séquences nucléotidiques par traduction inverse

Pour les procédures de vérification de la section 7.1.2, deux jeux de séquences nucléotidiques ont été construites respectivement sur la souris et l'homme.

Pour la souris, un jeu de 1751 protéines à été construit à partir des annotations Swissprot disponibles. Le but étant ici de pouvoir travailler aussi sur la séquence nucléotidique pouvant être associée à chacun des gènes, une étape de "traduction inverse" *in silico* a été faite de la manière suivante :

- `tblastp` de chacune des protéines contre la base de données d'ADN codant pleine longueur mises à disposition sur le site *Mouse Genome Informatics*[2].
- sélection des protéines présentant au moins 80% d'identité de séquence.

Le seuil d'identité de séquence a été fixé à 80 % afin de permettre une meilleure représentation de chaque compartiment et comporte 212 séquences cytoplasmiques, 269 nucléaires, 106 mitochondriales, 163 extracellulaires et 602 membranaires, totalisant 1382 protéines.

La même procédure a été effectuée sur le jeu de protéines humaines avec les ADNc pleine longueurs mis a disposition sur la "H-invitationnal database". En raison du grand

[2] `http ://www.informatics.jax.org/`

nombre d'ADNc disponibles sur cette base de données, le seuil d'identité de séquence minimal a été fixé à 90%. On a donc conservé les séquences nucléotidiques pour 200 protéines cytoplasmiques, 135 du réticulum endoplasmique, 150 mitochondriales et 157 nucléaires.

Chapitre 5

Etat de l'art

Ce chapitre présente une synthèse des différentes méthodes proposées sur le problème d'annotation du compartiment cellulaire d'une protéine. Notons qu'à l'exception des travaux présentés en 5.4, la plupart des méthodes extraient les descripteurs uniquement à partir de la séquence de la protéine.

Une partie des méthodes a pour objectif de modéliser les signaux d'adressage biologiquement identifiés. Par exemple la prédiction de l'adressage de la protéine dans la voie secrétoire ou la mitochondrie se fondera sur la détection d'un signal d'adressage à l'extrémité N-terminale. Les méthodes de modélisation des signaux, mises au point suivant le compartiment adressé seront présentées en section 5.1.

Remarquons que, dans le cas des signaux N-terminaux, la performance de ces stratégies pâtit généralement des incertitudes de prédiction pour le codon start. Par ailleurs, ces méthodes ne permettent de prédire que les protéines adressées dans certains compartiments et pourraient s'avérer trop conservatives. Ceci est le cas des signaux de localisation dans le noyau par exemple.

Ainsi, on présentera par la suite des méthodes alternatives qui utilisent d'autres types d'information : celles extraites de la séquence globale (5.2), telles que les fréquences en acides aminés ou les occurrences de motifs fonctionnels connus ; ou bien encore l'utilisation de l'information relative aux protéines homologues (5.3).

Enfin, quelques auteurs ont proposé une approche intégrative, où le développement d'un système expert combine le plus grand nombre d'informations disponibles pour la mise au point du classificateur. Les travaux publiés à ce sujet seront présentés en section (5.4).

Notons aussi que la plupart des méthodes présentées développent en parallèle des prédictions sur les protéines d'organismes procaryotes. Ces résultats ne seront pas présentés par la suite, mon travail s'étant focalisé sur les organismes eucaryotes.

5.1 Détection des signaux d'adressage

5.1.1 Méthodes de modélisation

La partie de séquence reconnue par le complexe responsable de l'adressage de la protéine possède des caractéristiques compositionnelles bien précises. Ainsi, plusieurs méthodes travaillant localement sur la séquence protéique ont été proposées.

La méthode de description la plus simple est celle utilisant une expression régulière, dans l'esprit de celles mises à disposition dans la banque de données prosite[1]. Par exemple, un des signaux d'adressage d'une protéine dans le noyau est décrit comme `KRKx{11}KKKSKK` qui correspond à la chaîne composée des deux motifs `KRK` et `KKKSKK` séparés de 11 lettres. Le signal d'adressage est ensuite détecté par l'occurence du motif dans la séquence.

Cependant, dans plusieurs cas, l'appariement spécifique entre le complexe protéique responsable de l'adressage et le signal à proprement parler correspond à un minimum énergétique. Une approche statistique sur la suite de lettre pourrait peut être mieux retranscrire ce phénomène.

Dans ce sens les premières approches proposaient l'utilisation d'une matrice positionelle pour décrire le site de fixation de la protéine. Si le site de fixation est de longueur k, sa loi d'apparition est décrite par k lois multinomiales $p_1, \ldots, p_k$. La probabilité d'une séquence $s_1 \ldots s_k$ d'être un site de fixation se calcule alors comme $p_1(s_1)p_2(s_2)\ldots p_k(s_k)$. La matrice positionnelle fait donc référence à la matrice à 20 lignes (taille de l'alphabet d'acides aminés) et k colonnes des $p_1 \ldots p_k$ mis côte à côte.

Cette méthode a ensuite été généralisée par la spécification de Chaînes de Markov cachées (CMC) modélisant la composition du site d'adressage. Dans ces cas la classification se fait ensuite par un rapport de vraisemblance entre un modèle estimé sur un jeu de séquences positives et un modèle représentant l'hypothèse nulle (estimé par exemple sur des séquences ne possédant pas de site de clivage). Plus de détails sur les modèles utilisés seront présentés dans les sections suivantes.

Dans plusieurs cas, l'utilisation de réseaux de neurones a produit des résultats de grande qualité. Ces réseaux, en général à plusieurs couches, permettent ainsi de prendre en entrée l'information concernant les résidus présents dans une fenêtre de longueur fixée, conjointement avec des informations concernant les propriétés physico-chimiques du contexte.

Enfin, en section 5.1.2.1 on présentera une approche qui propose différents recodages numériques des sous-séquences présentes autour du site pour appliquer des techniques de fouille de données.

[1] `http ://www.expasy.org/prosite/`

5.1.2 Résultats

5.1.2.1 Signaux d'adressage N-terminaux

L'analyse de cette catégorie de signaux impose de répondre à deux questions successives : la séquence possède-t-elle un signal d'adressage ? Si c'est le cas, à quelle position se trouve le site de clivage ?

Les premières méthodes proposées pour la détection des peptides signaux ont été développés il y a vingtaine d'années par [McG85] et [vH86] à l'aide de matrices positionnelles, la sensibilité avoisinant alors les 75-80 %.

Ensuite, avec l'augmentation des données expérimentales disponibles, ces méthodes ont été améliorées à l'aide de réseaux de neurones dans l'outil TargetP[2] présenté dans [ENBvH00]. Ce prédicteur combine des outils de prédiction pour le peptide signal (SP avec le programme signalP) et les signaux d'adressage dans la mitochondrie (mTP avec mitoP) ou le chloroplaste (cTP avec chloroP).

L'architecture générale de TargetP est présentée sur la figure 5.1. Le prédicteur combine ainsi deux couches de réseaux de neurones. La première couche est constituée de 3 ou 4 réseaux spécialisés pour la détection de chaque signal d'adressage. La seconde intègre les résultats des différents réseaux pour la prédiction à partir des résultats du réseau de neurones pour les 100 premières positions. Les auteurs notent que la seconde couche correspond presque à une séparation linéaire des classes. La taille des fenêtres a été fixée à 55 résidus pour le cTP, 35 pour le mTP et 31 (plant) ou 27 (eucaryotes non végétal) pour le SP. Bien entendu les jeux de séquences ont été réduit sur les séquences homologues pour les 65 premières positions.

Remarquons que le programme SignalP a été amélioré à deux reprises, une version 2, existant sous la forme d'un réseau de neurones ou d'une CMC, permettait déjà une meilleure détection des sites de clivage. La version CMC possédant l'avantage de mieux discriminer les ancres protéiques des signal-peptides. Récemment, [DBNvHB04] ont proposé une amélioration de SignalP, avec la version 3.0 en prenant en compte la composition en acides aminés pour la détection du site de clivage. Ainsi, la sensibilité de détection de la position du site de clivage augmente de 6 à 17 % suivant le règne. Le système de score proposé pour la détection du signal peptide a en outre été amélioré.

[KKS04] proposent quant à eux une extension de la méthode de SignalP-HMM, en permettant la détection conjointe du signal peptide et de la topologie de la protéine, si celle-ci est transmembranaire. La figure 5.2 présente l'architecture sur les états cachés mise au point pour le logiciel développé : Phobius[3]. Globalement, le modèle peut être vu comme la combinaison d'une CMC pour la détection des segments transmembranaire et la modélisation du signal peptide par SignalP-HMM (5.2(a)). Le sous-modèle du signal peptide (5.2(c)) est découpée suivant les régions n, h et c permettant de coller au mieux a la loi de longueur du signal peptide. Plus généralement, ce problème donne lieu à

[2] `http :/www.cbs.dtu.dk/services/TargetP/` [3] `http ://phobius.cgb.ki.se/`

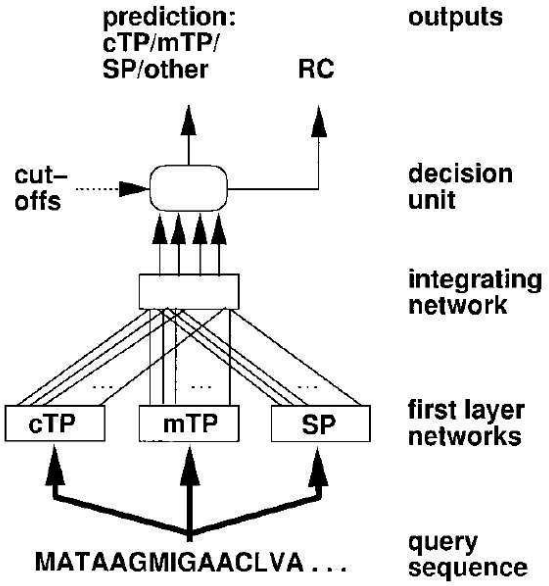


FIG. 5.1 – architecture de TargetP, tiré de [ENBvH00]. Le classificateur est construit à partir de deux couches de réseaux de neurones.

la mise au point d'un modèle extrêmement contraint sur la structure des états cachés. En comparant les performances du programme avec SignalP v2.0 pour la détection du signal peptide, phobius possède une sensibilité de l'ordre de 76% contre 82 % pour la version CMC de SignalP (85% pour les réseaux de neurones). Rappelons que l'intérêt majeur de Phobius est de détecter la topologie de la protéine, c'est à dire les segments transmembranaires et la position des coudes (cytoplasmique ou extracellulaire) et que ses performances sont supérieures au programme TMHMM pour la localisation des segments transmembranaires.

Enfin, le travail de [BTM+02] (logiciel iPSORT[4]) propose une méthode de recodage des séquences, soit numérique, par un index sur les acides aminés, soit par une réduction d'alphabet puis des recherches de motifs. Ainsi, les auteurs proposent une exploration de l'espace des recodages possibles des sous-segments de la séquence N-terminale, en combinant ensuite logiquement les règles les plus discriminantes. L'un des intérêt de cette approche très simple est qu'elle a permis l'extraction de règles pertinentes. Par exemple le système reconnaît la présence d'hélices α amphiphiles connues comme pouvant faire partie du signal d'adressage à la mitochondrie.

[4] `http ://hc.ims.u-tokyo.ac.jp/iPSORT`

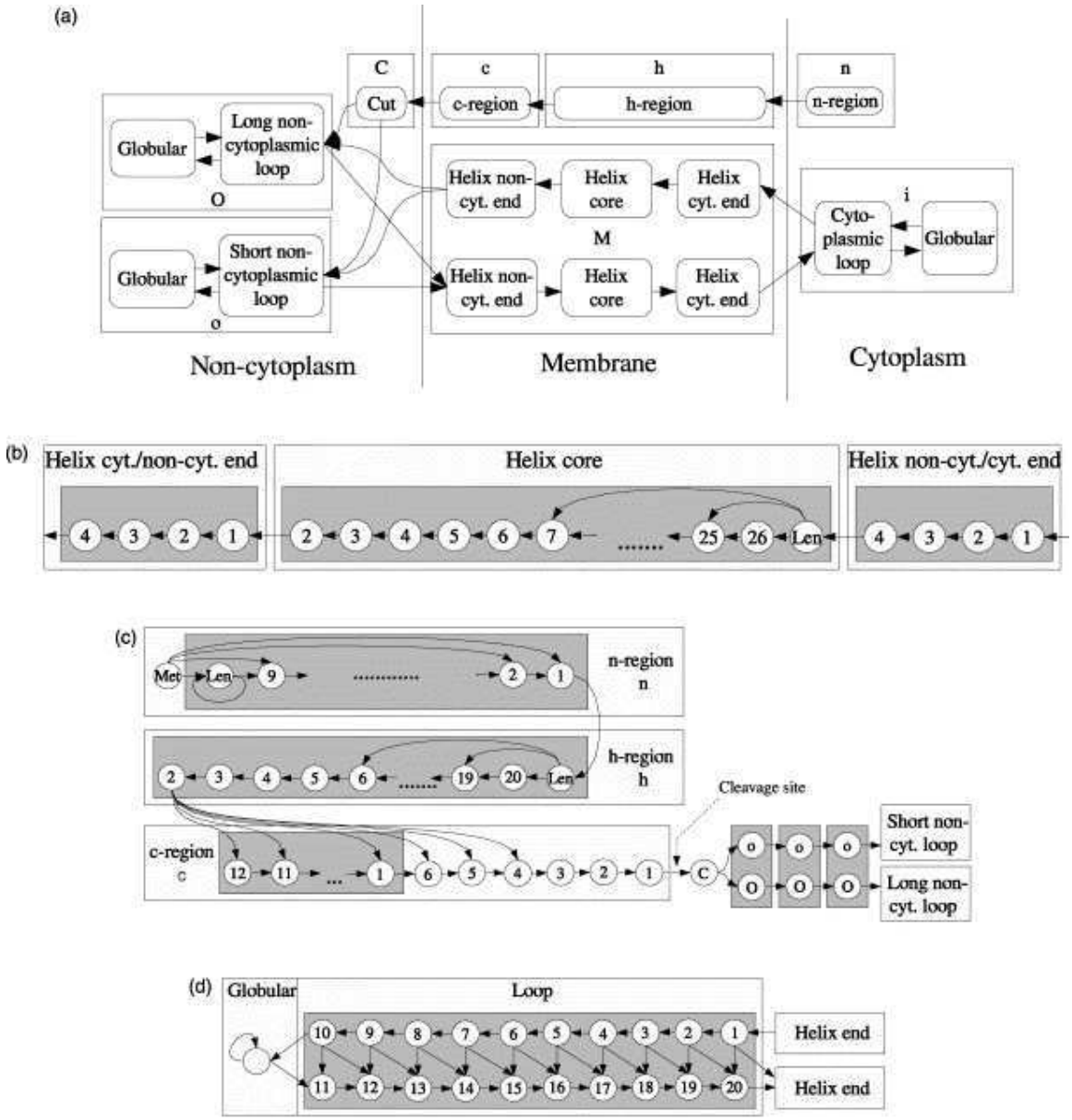


FIG. 5.2 – architecture de phobius, tiré de [KKS04]. Les boîtes grises correspondent à des états liés, *i.e.* possédant les mêmes lois d'émission des acides aminés (a) vue d'ensemble du modèle, (b) sous modèle pour les hélices transmembranaires, (c) sous modèle du peptide signal, (d) boucles cytoplasmiques et non-cytoplasmiques courtes.

5.1.2.2 Signal d'adressage au péroxisome de type 1

A l'heure actuelle, en raison de la quantité de données disponibles, les outils de prédiction proposent uniquement la détection d'un signal d'adressage de type 1 (PTS1).

	Signal	effectifs	TargetP		iPSORT	
			sensibilité	spécificité	sensibilité	spécificité
Plante	cTP	141	85	69	68	71
	mTP	368	82	90	84	86
	SP	269	91	96	91	98
	autre	162	85	78	83	70
	total	940	85,3		83,4	
Eucaryote non végétal	mTP	371	80	67	74	68
	SP	715	96	92	92	92
	autre	1652	88	97	90	92
	total	2738	90		88,5	

TAB. 5.1 – performance des méthodes de classification TargetP et iPSORT sur le jeu de données publié dans [ENBvH00]

Comme on l'a vu précédemment, ce signal est décrit pour la majorité des protéines par les trois derniers résidus C-terminaux `-SKL-`. Le motif PROSITE plus général est décrit comme `[ACGNST][HKR][AFILMVY]` mais peut mener à l'annotation d'un grand nombre de protéines n'étant pas péroxisomale (et donc une chute de spécificité).

En outre, Pex5, qui permet l'adressage des protéines dans le péroxisome doit pouvoir interagir avec celles-ci sur leur extrémité C-terminale. Cette contrainte d'interaction se traduit par certaines caractéristiques compositionnelles en amont du signal tripeptide. En effet, la figure 5.3 présente une comparaison de la quantité d'information par position pour des séquences péroxisomales et des séquences possédant un signal de type péroxisomal mais non adressées dans le péroxisome. Comme on peut le constater, les résidus présents sur les positions 3 à 12 en amont de l'extrémité C-terminale contiennent une information propre au site de fixation. Ainsi, pour les deux méthodes présentées, les auteurs utiliseront l'information compositionnelle ou physico-chimique présente sur une douzaine de résidus en amont du site de clivage.

Deux méthodes proposent actuellement la détection du signal PTS1 :

- le logiciel PeroxiP[5], présenté dans [EEvHC03], qui procède en deux étapes : un filtrage sur les protéines possédant le tripeptide présent en C-terminal puis une étape de classification supervisée sur les résidus présents dans la boîte -3 à -12 et la composition globale en acides aminés.
- Le prédicteur 'PTS1'[6] présenté dans [NMSE$^+$03b]. Les auteurs, partant d'une étude antérieure détaillée sur les propriétés d'interaction de la séquence avec Pex5 ou d'accessibilité au solvant ([NMSE$^+$03a]), proposent une combinaison linéaire de plusieurs matrices positionnelles sur les 12 résidus C-terminaux avec des pro-

[5] `http ://www.sbc.su.se/ arne/PeroxiP/` [6] `http ://mendel.imp.univie.ac.at/PTS1/`

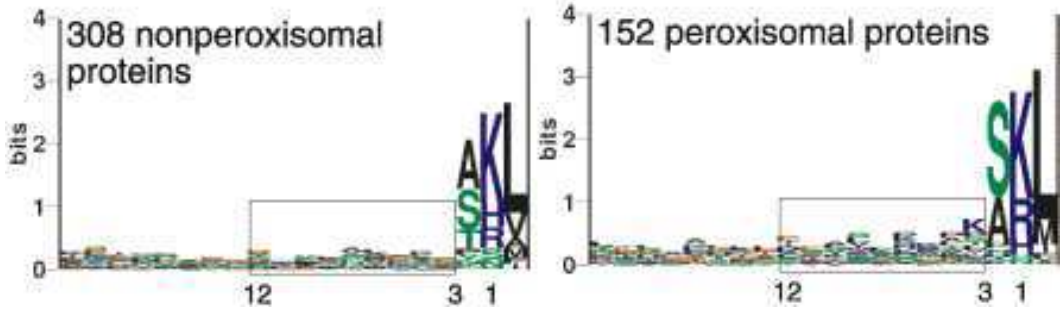


FIG. 5.3 – Tiré de [EEvHC03]. Comparatifs des logos de séquence pour la partie C-terminale de 152 protéines péroxisomales et 308 protéines non péroxisomale mais possédant un signal de type PTS1. Les boîtes de 9 résidus correspondent au contexte conservé pour l'utilisation du prédicteur par les auteurs.

priétés physico-chimiques globales sur ce fragment.

Détaillons rapidement les caractéristiques de ces deux méthodes avant de présenter leurs performances respectives.

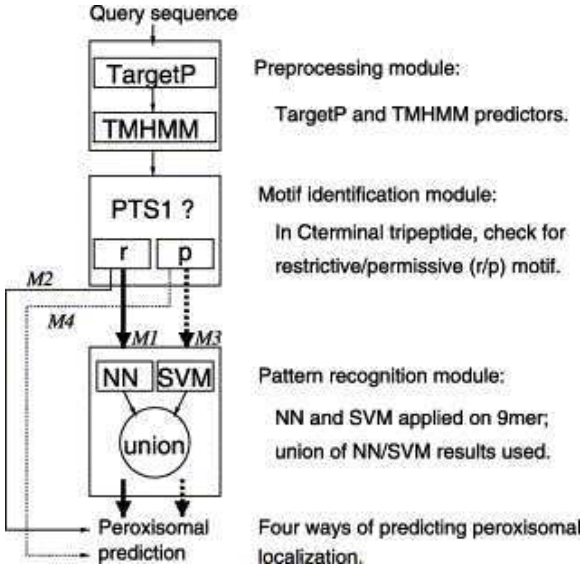


FIG. 5.4 – Tiré de [EEvHC03]. Etapes de prédiction du programme PeroxiP. Comme on peut le constater le signal PTS1 peut être détecté par 4 méthodes différentes.

PeroxiP, dont le processus de classification est résumé en figure 5.4, procède par deux étapes successives de filtrage. D'abord, les protéines prédites comme étant sécrétées ou transmembranaires sont éliminées. Cette étape est réalisée avec TargetP ou TMHMM. Ensuite, le tripeptide présent en C-terminal est vérifié pour correspondre à

l'un des 32 tripeptides présent sur le jeu d'apprentissage (motif (r)) ou à l'expression régulière construite à partir de ces mêmes séquences (motif (p)) :
`[ACHKNPST]-[HKNQRS]-[AFILMV]` et correspondant à 288 tripeptides possibles.

Ensuite, la protéine peut directement être prédite comme péroxisomale (méthodes M2 et M4) ou être soumise à une procédure de classification utilisant comme descripteurs les neuf résidus en amont du tripeptide et la composition globale en acides aminés. Pour la classification, les auteurs combinent par un OU logique les résultats par réseau de neurones et SVM (noyau polynomial) (méthodes M1 et M2 respectivement pour les motifs de type (r) ou (p)). Cette dernière étape est rajoutée afin d'augmenter la spécifité de la méthode.

Le prédicteur PTS1, utilisant les résultats parus dans [NMSE+03a], propose une prédiction spécifique sur trois types d'organismes : métazoaires, fungi et le reste.

Ils combinent leurs différentes constatations à l'aide de la fonction de score S suivante :

$$S = \underbrace{\sum_{i=1}^{3} S^{i}_{\text{profil}}}_{\text{propriétés de séquence}} + \underbrace{\sum_{i=1}^{22} S^{i}_{\text{ppt}}}_{\text{propriétés physico chimiques}}$$

Chacun des scores S^{i}_{profil} consiste alors en une combinaison de matrices positionnelles mettant l'accent sur les différentes contraintes identifiées pour la bonne reconnaissance de PTS1 : le tripeptide C-terminal (S^{3}_{profil}), les contraintes d'interaction avec Pex5, et l'accessibilité au solvant du site de fixation.

Les scores S^{i}_{ppt} sont introduits afin de pénaliser des séquences dont les propriétés physico-chimiques de certains résidus, identifiés pour être correlés à la présence du signal, s'écartent fortement des valeurs observées sur le jeu d'apprentissage. Par exemple, S^{18}_{ppt} rajoute une pénalité dépendant de la flexibilité de la chaîne entre les positions -6 et -1.

Les performances des deux méthodes sont résumées dans la table 5.2. On peut voir que le prédicteur PTS1 semble posséder de meilleures performances. Ceci peut s'expliquer par le soin particulier apporté par les auteurs à la mise au point d'une fonction de score reflétant au mieux les contraintes énergétiques et chimiques nécessaires au bon appariement de Pex5 et du du signal PTS1. Cependant, on notera que le classificateur construit par PTS1 a nécessité l'ajustement d'un grand nombre de paramètres (au total près d'une quarantaine de coefficients) et correspond donc à une approche assez conservative. A l'inverse PeroxiP serait plus facile à mettre à jour avec l'identification de nouvelles séquences péroxisomales.

Remarquons aussi que les auteurs de PTS1 font une validation intéressante en testant la validité biologique de leur prédicteur sur des expériences publiées testant l'influence de mutagénèses dirigées pour l'import de protéines dans le péroxisome. Sur 88 protéines prises en compte, les auteurs reportent une sensibilité de 70%, et une spécificité

à 98,4%.

	PeroxiP	PTS1
Jeux de données avec signal	152 séquences de Swissprot	•105 séquences par double hybride •205 séquences dans SWALL
Jeux de données sans signal	308 protéines non péroxisomales possédant le signal dégénéré	20544 protéines procaryotes
Sensibilité	78 % (50%)	84,2 %
Spécificité	64 % (64%)	81 %

TAB. 5.2 – Résultats de classifications pour PeroxiP et PTS1. Les résultats de classification donnés entre parenthèses pour PeroxiP correspondent à un test sur le jeu de protéines humaines possédant une localisation cellulaire annotée dans Swissprot (5174 protéines dont 28 péroxisomales).

5.1.2.3 Signaux de localisation nucléaire

Les signaux de localisation nucléaires (NLS) annotés expérimentalement sont connus comme pouvant être composés d'un ou de deux motifs. La plupart des motifs simples sont caractérisés par une suite d'acides aminés chargés positivement. Les signaux composés de deux motifs correspondent eux à deux séquences d'acides aminés basiques séparés par 9 à 12 acides aminés. Un des problèmes majeurs de détection de ces signaux est leur possibilité d'être localisés n'importe où sur la séquence, la contrainte étant leur accessibilité à la protéine d'import dans le noyau.

Pour l'heure, une seule approche a été proposée par [CNR00] qui, en rassemblant 114 NLS annotés expérimentalement dans la littérature, ont augmenté le jeu de motifs à 308 par recherche d'homologues et "mutagénèse *in silico*".

En effet, les NLS actuellement identifiés ne prennent en compte que 10 % des protéines nucléaires connues. Afin de remédier à ce problème, les auteurs ont d'abord étendu le jeu de motifs par la détection d'homologues dans SWISSPROT v. 38.0 et possédant au moins 80% d'identité de séquence. Ce jeu de motifs a été étendu à nouveau en leur appliquant les mutations, insertions ou délétions augmentant le nombre de protéines nucléaires prises en compte. Ensuite, tous les motifs générant au moins un faux positif ou présents dans une seule famille de protéines nucléaires ont été retirés. Ainsi, les auteurs atteignent une sensibilité de 43% en maintenant une spécificité de 100%. Ce jeu de motifs est compilé dans la base de données NLSdb[7] accessible par SRS (Sequence Retrieval System) ([NCR03]).

[7] `http ://cubic.bioc.columbia.edu/db/NLSdb/`

5.2 Détection à partir d'informations globales

Dans cette section, nous ne décrirons que les méthodes qui nous ont semblé originales, soit de par la spécificité des jeux de données ou des classificateurs, soit sur les descripteurs extraits de la séquence.

5.2.1 Composition en acides aminés

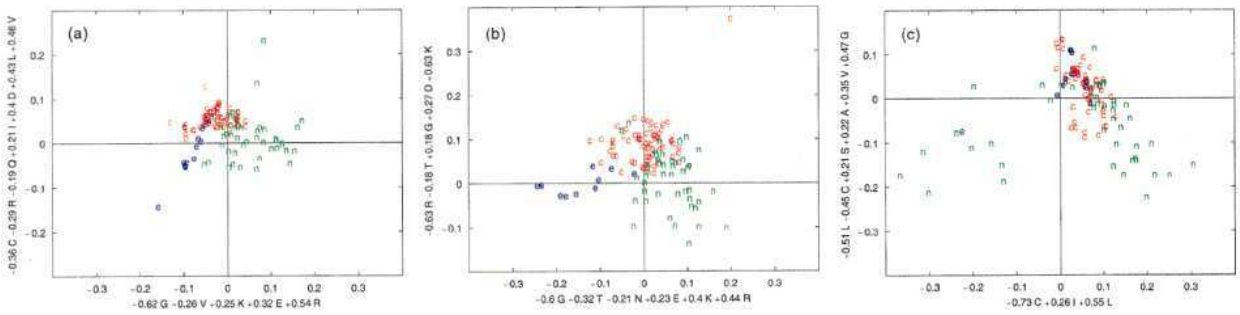


FIG. 5.5 – Tiré de [AOR98]. Deux premiers axes de l'analyse en composantes principale pour la composition en acides aminés (a) globale, (b) accessibles au solvant et (c) enfouis dans le coeur de la protéine. rouge : cytoplasme, vert : noyau et bleu extracellulaire.

Reprenant la première étude de [NN92], la figure 5.5 montre les deux premiers axes d'une analyse en composante principale sur les fréquences des acides aminés pour 3 compartiments (tiré de [AOR98]). Comme on peut le constater la composition moyenne en acides aminés des protéines tend à séparer les protéines des 3 compartiments. Ce qui peut laisser penser que la composition en acides aminés est une information pertinente (un descripteur informatif) pour prédire la localisation cellulaire des protéines. La plupart des méthodes présentées dans cette section utilisent ainsi les variations de fréquences d'acides aminés pour prédire à quel compartiment la protéine est finalement adressée. Remarquons aussi une étude de [AOR98] sur des protéines de structure connue localisées dans trois compartiments, et montrant que les fréquences des résidus les plus exposés au solvant étaient plus corrélées avec la localisation cellulaire de la protéine. Suivant ces constations, le système LOC3Dini[8] a été développé par [NR03] pour la prédiction du compartiment cellulaire de protéines dont la structure 3D est connue. LOC3Dini utilise comme descripteurs la composition en acides aminés globale, la composition pour les résidus accessibles au solvant, et les compositions conditionnelles à la structure secondaire (hélice, feuillet, coude).

[8] `http :/cubic.loc.columbia.edu/db/LOC3D`

Partant du travail initial de [CAPPE97] (qui classait chaque protéine par analyse discriminante linéaire parmi quatre compartiments, sur la base de leur composition en acides aminés), plusieurs auteurs ont proposé une stratégie de classification différente à partir des mêmes descripteurs. [RH98] proposent ainsi une méthode par réseaux de neurones sur 5 compartiments pour tous les organismes eucaryotes exceptés les plantes. Leur méthode atteint une sensibilité globale de 66% (logiciel NNPSL[9]). [HS01] utilisent eux des SVM à noyaux gaussiens pour 5 compartiments et atteignent une sensibilité de 79% avec leur outil SubLoc[10].

Cependant, comme on le verra dans la suite (table 5.3), les performances de ces méthodes ont été fortement surestimées, sans doute du fait qu'il reste dans ces jeux des protéines possédant plus de 90% de similarité de séquence.

[Cho01] propose d'améliorer la qualité des prédictions en introduisant des indicateurs sur l'ordre d'apparition des lettres dans la séquence. Il propose ainsi de rajouter aux 20 compositions en acides aminés, k termes θ_1 à θ_k prenant en compte la corrélation globale en terme de caractéristiques physico-chimiques entre les résidus séparés de l lettres. Ainsi sa "formule" pour une protéine de longueur ℓ, $(y_1, \ldots, y_\ell)$ est :

$$\theta_\lambda = \frac{1}{L-\lambda} \sum_{i=1}^{\ell-\lambda} \Theta(y_i, y_{i+\lambda})$$

$$\text{avec} \quad \Theta(y_i, y_j) = \frac{1}{3}\left(\underbrace{(\sigma_1(y_i) - \sigma_1(y_j))^2}_{\text{hydrophobie}} + \underbrace{(\sigma_2(y_i) - \sigma_2(y_j))^2}_{\text{hydropathie}} + \underbrace{((\sigma_3(y_i) - \sigma_3(y_j))^2}_{\text{masse}} \right),$$

où les σ_i correspondent aux valeurs centrées réduites pour les propriétés physico-chimiques indiquées. Le nombre de termes k à intégrer au vecteur de composition est déterminé par optimisation de la sensibilité sur le jeu de données (dans [Cho01], la valeur $k = 13$ est sélectionnée). Leur méthode de classification porte sur 12 compartiments et utilise une analyse discriminante quadratique [KCC03] ou des SVM [YDXJXbKC02] suivant la publication. Leur résultats de classification seront présentés plus loin, table 5.4.

Dans le même esprit, [Yua99], propose une simplification des modèles de Markov pour la prédiction du compartiment cellulaire en travaillant sur le jeu de données mis au point par [RH98]. La sensibilité de la méthode à l'aide de modèles d'ordre 1 atteint 70 % par procédure LOO. Afin d'améliorer cette sensibilité l'auteur choisi de prendre en compte l'information des résidus adjacents mais en limitant linéairement le nombre de paramètres devant être estimés. L'auteur propose ainsi d'écrire la dépendance sur k positions de la manière suivante :

$$\mathbf{P}(Y_t = y_t \mid Y_{t-1} = y_{t-1} \ldots Y_1 = y_1) = \prod_{i=1}^{k} \mathbf{P}(Y_t = y_t \mid Y_{t-i} = y_{t-i}) = \prod_{i=1}^{k} \Pi_i(y_{t-i}, y_t)$$

[9] `http ://predict.sanger.ac.uk/nnpsl` (le serveur ne semble plus accessible).
[10] `http ://www.bioinfo.tsinghua.edu.cn/SubLoc/`

En fixant $k = 4$, les auteurs obtiennent ainsi une sensibilité globale de 73%. (78% cytoplasme, 62% extracellulaire, 74% nucléaire et 69% mitochondrie).

Enfin, récemment [NR05] ont proposé une méthode de classification par arbre dans le même esprit que celle présentée en section 6.3. Le système développé, LOCTree[11] propose la classification d'une séquence dans 6 (plantes) ou 5 (tous eucaryotes à l'exception des plantes) compartiments. Le système décisionnel implémente un arbre reproduisant les mécanismes d'adressage d'une protéine. Il est constitué d'un classificateur par SVM à chaque noeud. Les auteurs utilisent comme informations : les compositions en acides aminés, la composition des 50 premiers résidus et la composition dans les 3 types de structures secondaires, donc 100 descripteurs différents. Avant de présenter les résultats qu'ils obtiennent et la comparaison avec les autres méthodes, notons que le jeu proposé par ces auteurs est le seul où la redondance en protéines homologues a été réduite en imposant au plus 25% d'identité de séquence sur un alignement de 250 résidus. Le jeu de validation ayant été construit à partir de SWISSPROT version 40, les auteurs ont utilisé SWISSPROT version 41 pour la seule comparaison disponible à ce jour avec d'autres méthodes (en réduisant à nouveau le jeu sur les protéines pouvant être homologues avec celles de SWISSPROT version 40). Remarquons que les auteurs ont augmenté leurs jeux de données à l'aide des méthodes LOCHom et LOCKey (voir 5.3 pour plus de précisions), ceci a permis un gain en sensibilité de plus de 7%.

La table 5.3 ci-dessus compare les résultats obtenus entre les dernières méthodes en date pour le problème de la localisation avec un nombre limité de compartiments. Ceci amène à penser que le classificateur LOCTree est à l'heure actuelle le plus performant pour la prédiction de la localisation pour un nombre réduit de compartiments en prenant en compte les informations globales sur la séquence. En comparant les performances obtenues sur le jeu de test déduit de swissprot 41 avec celles publiées, il apparaît que celles-ci, au moins pour ces classificateurs, ont été surestimées. Ceci peut être entre autre expliqué par le fait que les jeux de donnés construits pour la validation, n'ont pas assez été filtré sur l'homologie. En effet, sur le compartiment cytoplasmique, les performances publiées pour SubLoc et NNPSL surestiment de 30% la sensibilité observée sur le jeu de test indépendant. En outre, ces différences sont trop marquées pour pouvoir être attribuées à de trop faibles effectifs lors de l'entraînement de ces classificateurs. Etonnamment, la méthode LOCTree présente de meilleures sensibilités et spécificités sur le jeu de test pour tous les compartiments. La sensibilité de LOCTree augmente aussi significativement pour les protéines cytoplasmiques entre leur jeu publié et leur jeu de test. Ceci pourrait être expliqué par certains biais propres à ce jeu de données, ou par le fait que les effectifs pour certains compartiments sont trop faibles pour permettre un bonne estimation de la spécificité (par exemple la voie sécrétoire et la mitochondrie contiennent moins d'une cinquantaine de protéines).

Pour clore cette section, présentons maintenant les différentes méthodes ayant été

[11] `http ://rostlab.org/services/LOCtree/`

		LOCtree		TargetP		SubLoc		PSORT II	NNPSL	
		publié	test	publié	test	publié	test	test	publié	test
Voie Sécrétoire	sens.	81	87	96	93					
	spec.	80	90	92	73					
Extracell.	sens.	83	86			80	73	91	75	62
	spec.	81	93				53	32		63
Noyau	sens.	78	77			87	64	56	72	67
	spec.	78	85				71	75		59
Cytoplasme	sens.	63	82			77	43	47	55	42
	spec.	66	64				56	47		38
Mitochondrie	sens.	70	73	80	54	56	48	46	61	30
	spec.	67	78	67	75		59	59		67
Sensibilité globale		74	78			79	57	51	66	52

TAB. 5.3 – Comparatif des résultats de classification pour LOCTree, TargetP, SubLOC, PSORTII et NNPSL. Pour chaque méthode, les résultats publiés dans les articles correspondants ont été mis dans la colonne publié. La colonne test correspond aux résultats présentés dans [NR05] sur les protéines de SWISSPROT version 41 ne possédant aucun homologue dans SWISSPROT 40. Les résultats publiés de PSORT II n'ont pas été intégrés. Remarquons que pour son jeu d'apprentissage, LOCTree intègre les protéines péroxisomales dans la voie sécrétoire.

proposées pour construire un classificateur prédisant la localisation cellulaire sur un plus grand nombre de compartiments. La plupart des méthodes publiées proposent une classification sur 12 compartiments, tous organismes eucaryotes confondus. Même si dans le cas de Park et Kanehisa, le classificateur proposé sur leur site internet a été spécialisé pour les animaux ou les plantes, nous ne présenterons ici que le classificateur général afin de simplifier les comparaisons.

[PK03] ont proposé une méthode utilisant comme information la composition en acides aminés et en doublets d'acides aminés, séparés de 1 à 3 positions. Un SVM à noyau gaussien est ensuite entraîné pour chaque type de descripteur, avec une stratégie de 1-vs-all (cf. section 6.2). La classe est ensuite attribuée par vote sur les différents SVM entraînés. Remarquons que cette approche peut directement être mise en relation avec le travail de [Yua99] comme présenté en section 3.2.

Finalement, [SCLK05] ont proposé une manière alternative pour prendre en compte les influences de l'ordre d'apparition des lettres dans la séquence, en utilisant les propriétés physico-chimiques des acides aminés. Afin de pouvoir appliquer la stratégie par SVM à des individus de même longueur, les auteurs segmentent le jeu de données en quatre groupes de longueurs $L = (50, 150, 300, 450)$. Ensuite, chaque protéine de longueur ℓ est ramenée à la première longueur qu'elle majore L en moyennant la propriété

physico-chimique considérée sur autant d'acides aminés que nécessaires par groupes de $[\ell/L] + 1$ résidus. Cette procédure de lissage commence au début de la séquence[12]. Les indices physico-chimiques pris en compte sont déterminés a partir de la base de données AAindex[13] par évaluation de la performance pour chaque indice et couple de compartiments : pour chaque couple de compartiments, les cinq meilleurs indices sont sélectionnés pour être intégrés ensuite dans le SVM qui fera la classification. De manière intéressante, sur les 75 indices sélectionnés (6 compartiments et 5 indices par compartiment et sans que les auteurs en fasse la constation), 12 (16%) sont tirées des premières publications sur l'influence de la composition **globale** en acide aminés sur la localisation cellulaire des protéines : [NN92], [NK92], et [CAPPE97]. Une grande partie des autres indices est liée à des propriétés structurales ou énergétiques des acides aminés.

Subcellular	Chou & Cai	P & K	pSLIP			
Location	LOO-CV	5-fold	5-fold		10-fold	
	sens.	sens.	sens.	spec.	sens.	spec.
Chloroplast	57	72	85	90	92	94
Cytoplasmic	88	72	84	86	88	92
Cytoskeleton	44	59	-	-	-	-
ER	31	47	-	-	-	-
Extracellular	57	78	92	96	94	98
Golgi Apparatus	12	15	-	-	-	-
Lysosomal	54	62	-	-	-	-
Mitochondrial	42	57	87	77	94	86
Nuclear	73	90	90	89	93	93
Peroxisomal	4	26	-	-	-	-
Plasma Membrane	91	92	94	95	97	95
Vacuolar	25	25	-	-	-	-
TA	75	78	90		93	
LA	48	58	89		93	

TAB. 5.4 – Tiré de [SCLK05]. Comparatifs des résultats de classification pour les méthodes portant sur 12 compartiments. Notons que les auteurs de pSLIP ont réduit le problème à 6 compartiments en raison de la baisse d'effectifs suite au regroupement par longueurs de séquence.

Leurs résultats sur le jeu de Park & Kanehisa (détaillé en section 4.1.2) sont présen-

[12] article disponible en ligne : `http ://www.biomedcentral.com/1471-2105/6/152` pour plus de précisions [13] `http ://www.genome.ad.jp/dbget/aaindex.html`

tés en table 5.4. Notons que si le jeu de Park & Kanehisa est nettoyé sur la similarité de séquences, Chou et Cai ont extrait leur jeu de SWISSPROT version 38 en retirant simplement les doublons présents, ce qui, comme on a pu le voir sur la table 5.3, biaise fortement l'estimation de la performance du classificateur (a fortiori pour des résultats obtenus par Leave One Out).

Ainsi, la méthode de Park est pour l'instant la plus performante pour plus de 6 compartiments. Les performances de pSLIP tiennent la comparaison les résultats présentés pour LOCTree. Deux points méritent cependant d'être soulignés : la procédure de nettoyage sur les protéines homologues est beaucoup plus stringente dans le cas de LOCTree, et cela aurait pu à nouveau biaiser favorablement les résultats présentés pour pSLIP.

En outre, pSLIP n'apporte pas d'information sur les protéines adressées dans la voie sécrétoire, et manque donc un petit peu de pertinence biologique.

5.2.2 Occurrence des motifs protéiques

Si la prédiction du compartiment cellulaire d'une protéine permet la détermination de groupes de protéines interagissant entre elles, ou participant de la même réaction biochimique, l'augmentation et l'amélioration des outils de détection automatique de motifs ou de domaines protéiques pourraient permettre l'inverse.

Ainsi, [MSBP02] ont proposé les premiers une étude descriptive portant sur 300 domaines SMART[14] définis comme associés à certains compartiments. Ensuite, par analyse des co-occurrences de ces motifs au sein des protéines de la base de données SP-TrEMBL (21 % des protéines possède un motif SMART), les auteurs en déduisent un graphe valué dont la projection est représentée sur la figure 5.6. Comme on peut le constater, les co-occurrences de domaines sont fortement corrélées à leur localisation cellulaire. Avec cette approche projective, les auteurs peuvent ainsi classer 284 des 300 domaines étudiés (95%) sans ambiguïté sur la localisation.

Cependant, si cette approche descriptive met bien en avant les liens entre la localisation cellulaire d'une protéine et les domaines d'interaction qu'elle porte, la méthode de classification en elle-même pourrait être améliorée. En effet, [CC02] proposent par exemple de simplement recoder une protéine comme le vecteur des occurrences des motifs présents dans la séquence. Bien que travaillant avec des vecteurs de grandes dimensions (le nombre de motifs possibles, soit 4092 dans ce cas), l'utilisation de SVM permet d'obtenir des performances raisonnables sur le jeu de données déjà utilisé par les auteurs (67% de sensibilité globale).

[STH04] propose une approche bayésienne permettant la classification de protéines humaines. Leur méthode (logiciel PSLT[15]) part de motifs appartenant à la base de

[14] `http ://smart.embl-heidelberg.de/` : une base de données de domaines protéiques.
[15] `http ://www.mcb.mcgill.ca/ hera/PSLT`

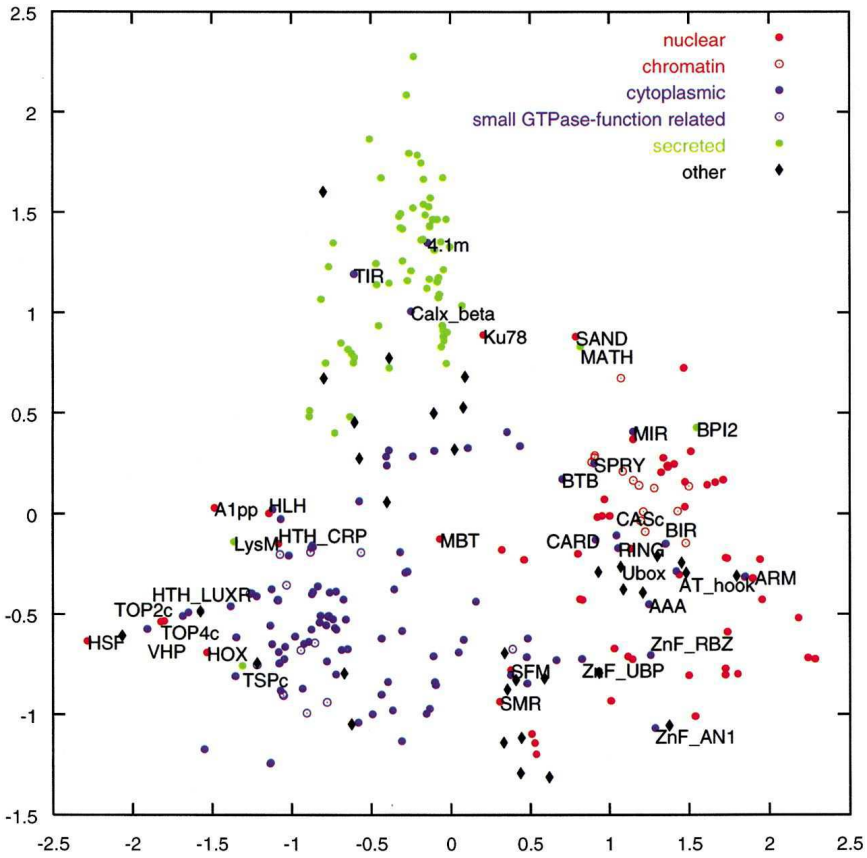


FIG. 5.6 – Tiré de [MSBP02]. Projection des 300 domaines protéiques SMART, coloriés suivant la localisation cellulaire associé à chaque domaine. Les axes correspondent au deux premières composantes principales dans la métrique construite sur les co-occurrences des domaines entre protéines. Les cercles creux identifient les domaines nucléaires liés à la chromatine ou ceux régulant des fonctions GTPase de type Ras. Les domaines annotés comme "autres" réfèrent à des domaines ne discriminant pas nécessairement une localisation cellulaire.

données Interpro[16], du nombre de segments transmembranaire (déterminés par l'outil TMHMM) et du peptide signal (prédit par le programme SignalP). Si un jeu de motifs $\mathcal{M}_\wp = \{M_1, \ldots, M_n\}$ est détecté dans la protéine $\wp$, pour chaque compartiment C, on calcule simplement sa probabilité a posteriori d'appartenir à un compartiment par la loi de Bayes :

$$\mathbf{P}(C \mid \mathcal{M}_\wp) = \mathbf{P}(\mathcal{M}_\wp \mid C)\frac{\mathbf{P}(C)}{\mathbf{P}(\mathcal{M}_\wp)}$$

où la probabilité a priori pour un jeu de motifs $\mathbf{P}(\mathcal{M}_\wp)$ est estimée sur toutes les pro-

[16] http ://www.ebi.ac.uk/interpro/

téines humaines de SWISSPROT version 41.25. Les probabilités a priori pour chaque compartiment $\mathbf{P}(C)$ ont été sélectionnées afin d'optimiser la sensibilité globale par algorithmes génétiques. Afin de prendre en compte le plus possible les co-occurrences de motifs, les probabilités des jeux de motifs conditionnellement à chaque compartiment sont estimées pour chaque groupement possible de motifs.

Les résultats obtenus par validation croisée pour cette méthode sont présentés dans la table 5.5. Comme on peut le constater, conditionnellement à la présence de motifs Interpro (74 % des protéines), la méthode possède une bonne sensibilité, et justifie ainsi de manière générale les approches par motif pour la prédiction de la localisation cellulaire. En outre, l'approche bayésienne permet d'avoir une information sur les protéines pouvant être multicompartimentales. Ainsi, en annotant la localisation cellulaire pour les 9793 protéines de SWISSPROT version 41.25, les auteurs estiment que 16% des protéines sont multicompartimentales. De manière intéressante, l'examen approfondi de l'annotation Swissprot de ces protéines a permis de confirmer leur caractère multicompartimentale.

Comme on pourra le constater dans la section suivante, l'utilisation d'informations fonctionnelles sur la séquence produit les méthodes les plus performantes à ce jour.

	couverture	validation croisée	
		sens.	spec.
ER	83.6%	69	83
Golgi	75.0%	60	74
Cytosol	89.4%	65	68
Nucleus	93.2%	93	84
Peroxisome	86.4%	43	50
Plasma membrane	94.1%	89	77
Lysosome	86.8%	60	71
Mitochondria	80.6%	67	61
Extracellular	90.1%	89	87
Sensiblité globale		78	
Couverture		74	

Tab. 5.5 – résultats obtenus par validation croisée 10 fois, tiré de [STH04]. Les sensibilités sont données pour les protéines possédant au moins un motif Interpro. La colonne couverture donne pour chaque compartiment la proportion de protéines possédant au moins un motif Interpro.

5.3 Méthodes fondées sur l'homologie

La méthode de transfert par homologie est souvent utilisée, qui pour modéliser la structure possible d'une protéine, qui pour inférer sur sa fonction. Cependant, suivant le problème considéré, le pourcentage de résidus identiques ou l'utilisation de scores statistiques (E-value de BLAST par exemple) peuvent influencer plus ou moins fortement les performances. [NR02b] ont évalué l'influence de différentes mesures de similarité sur la performance de leur méthode de prédiction qui se fonde sur l'homologie. Ils comparent ainsi la qualité de discrimination obtenue en seuillant sur le pourcentage d'identité de séquence, la E-value ou la mesure HSSP, développée pour détecter des homologues structurels. Pour une valeur HSSP supérieure à 5, les auteurs rapportent une spécificité supérieure à 80% pour les compartiments considérés. Néanmoins la sensibilité globale de la méthode ne dépasse pas 40% à ce seuil, sans doute de par le manque d'homologues. En effet, l'application de cette méthode à la base de données SWISSPROT version 37 leur permet d'annoter la localisation cellulaire de 24% des protéines[17]. Il est toutefois intéressant de voir que cette méthode fondée sur l'alignement global des protéines et ne tenant donc pas compte des différents signaux d'adressage, possède malgré tout de bonnes performances.

Afin d'augmenter le nombre de protéines prédictibles (en d'autres termes la couverture) [NR02a] ont développé LOCkey[18] qui classe une séquence sur la base des mots-clefs SWISSPROT de ses homologues. Les mots-clefs appropriés sont sélectionnés durant la phase d'entraînement par un critère d'entropie. Les auteurs obtiennent une sensibilité globale de 81,5% avec une couverture de 37%. Notons que ces résultats sont donnés par validation croisée pour un jeu nettoyé des homologues. Ainsi, la classification d'une protéine se fait uniquement sur la base des mots-clefs extraits de ses homologues, et non pas sur des homologues communs dans les jeux d'entraînement et de test.

[LSG+04] ont proposé une extension générale de ce système, avec l'outil Proteome Analyst spécialisé dans la prédiction de la localisation cellulaire (PA-SUB[19]) qui fonctionne sur 5 types d'organismes (animaux, plantes, champignons, bactéries Gram positive et négative). La classification d'une protéine se fait donc suivant les étapes suivantes :

- Recherche des protéines homologues par PSI-BLAST (une itération, 3 meilleurs "hits")
- extraction des attributs annotés dans SWISSPROT (parmis les mots-clefs, les motifs Interpro et les annotations de localisation cellulaire).
- Classification bayésienne simple pour le règne considéré sur les attributs extraits.

Le classificateur bayésien est entraîné en supposant les différents attributs indépen-

[17] disponible sur la base de données LOChom : `http ://cubic.bioc.columbia.edu/db/LOChom/`
[18] `http ://cubic.bioc.columbia.edu/db/LOCkey/` [19] `http ://www.cs.ualberta.ca/ bioinfo/PA/Sub/`

dants. Nommant $a_1, \ldots, a_n$ les attributs extraits d'une protéine $\wp$, la log-probabilité a posteriori pour chaque compartiment c se calculant comme :

$$\log \mathbf{P}(C_\wp = c \mid a_1, \ldots, a_n) = \mathbf{P}(C_\wp = c) + \sum_{i=1}^{n} \log \mathbf{P}(a_i \mid C_\wp = c) + \kappa,$$

Les compositions a priori pour chaque compartiment sont simplement estimées sur les données d'entraînement. La protéine est ensuite classée dans le compartiment le plus probable a posteriori. Cette formulation simple permet ainsi à l'utilisateur du programme d'obtenir une indication sur la contribution de chaque attribut à la classification. Notons que le nombre d'attributs différents, extraits de SWISSPROT peut être très grand et nuire à la qualité de la prédiction. Les auteurs procèdent donc à une étape de sélection de variables sur les attributs possédant une faible entropie relative par rapport aux compartiments.

La table 5.6 montre les performances de PA-SUB sur le jeu de données animales, et un test sur un unique organisme : le boeuf. Comme on peut le constater, ce classificateur semble le plus performant à l'heure actuelle. Cependant, du fait de la méthode par homologie, deux points restent à préciser :

- lors de l'évaluation des performances par validation croisée, les auteurs se contentent de retirer du jeu d'entraînement les protéines qui serviront au test. Il est donc possible que chaque protéine du jeu d'entraînement ait en commun un grand nombre d'homologues avec le jeu de test, ce qui pourrait biaiser favorablement les résultats. Quoi qu'il en soit, la méthode est aussi comparée avec LOCkey et possède une sensibilité de 93% sur les données à 34% de couverture (soit 10% de plus). Sur le jeu de données complètes, PA-SUB réalise 88% de sensibilité pour une couverture complète. Ceci peut être expliqué par le fait que : (1) le classifieur bayésien général possède de meilleures caractéristiques, (2) PA-SUB extrait les annotations interpro de chaque protéine.
- Quelle couverture peut-on attendre sur un organisme complet nouvellement séquencé ? Les auteurs reportent des couvertures de 74,5% sur *M. Musculus*, 60% sur *A. Thaliana* et 78,7% sur *S. Pombe*.

5.4 Systèmes “experts”

De manière intéressante, le premier logiciel proposant la prédiction de la localisation cellulaire d'une protéine parmi plusieurs compartiments était aussi un système expert. PSORT [NK92], intégrait alors un ensemble de règles “si-alors-sinon” pour la prédiction de la localisation cellulaire d'une protéine parmi 14 compartiments pour les animaux et 17 pour les plantes. Ensuite le système a ensuite été amélioré par [HN96] puis [NH99], en intégrant des schémas de classification par arbre de décision puis par

Compartiment	validation croisée			1-organisme : boeuf		
	num.	spec.	sens.	num.	spec.	sens.
noyau	2846	0.979	0.905	47	1.000	0.894
mitochondrie	1194	0.973	0.970	145	0.972	0.952
cytoplasme	1845	0.866	0.919	84	0.878	0.940
extracellulaire	3943	0.972	0.927	197	0.974	0.964
golgi	167	0.723	0.892	7	0.667	0.857
péroxisome	103	0.909	0.971	4	0.800	1.000
Rét. End.	457	0.868	0.952	14	0.824	1.000
lysosome	170	0.861	0.947	12	0.857	1.000
membrane	4820	0.957	0.938	218	0.966	0.917
Overall	15549	0.946	0.929	728	0.950	0.941

TAB. 5.6 – tiré de [LSG+04]. Résultats de classification pour PA-SUB sur les animaux et test sur un organisme (le boeuf). Pour le test, toutes les protéines du boeuf ont été retirées du jeu d'entraînement.

méthode des k plus proches voisins. La méthode proposait alors de classer les protéines de la levure parmi 10 compartiments, la sensibilité globale étant de 54%.

Reprenant une partie des attributs de séquence extraits par PSORT, [DG00] ont développé un prédicteur bayésien prédisant la localisation des protéines de la levure parmi 5 compartiments. De manière originale, les auteurs intégraient dans leur système des données d'expression (taux d'ARN messagers quantifiés par puce à ADNc), et les résultats d'expérience de knock-out[20] ou des motifs protéiques.

Récemment, les informations de PSORT et iPSORT ainsi que des propriétés de corrélation des acides aminés pour un jeu de données construit à partir d'Uniprot ont été intégrées dans le serveur WoLFPSORT[21]. La classification des protéines se fait par homologie ou par la méthode des k plus proches voisins.

On peut remarquer que ces systèmes experts sont peu utilisés. En effet, un nombre croissant de méthodes combinent différents types d'information (comme par exemple les occurrences de motifs ou l'homologie) et pourraient donc être apparentés aux susdits systèmes experts. Cependant, étant définis dans un cadre d'apprentissage statistique précis, il présentent l'avantage d'être facilement adaptables à l'apparition de nouvelles données. A l'inverse des classificateurs comme PSORT, mis au point par une approche descriptive, peuvent tendre à être trop conservatifs.

[20] viabilité de la cellule après délétion du gène. [21] `http ://wolfpsort.cbrc.jp/`

Bibliographie

[AOR98] M.A. Andrade, S.I. O'Donoghue, and B. Rost. Adaptation of protein surfaces to subcellular location. *Journal of Molecular Biology*, 276(517-525), 1998.

[BTM+02] Hideo Bannai, Yoshinori Tamada, Osamu Maruyama, Kenta Nakai, and Satoru Miyano. Extensive feature detection of N-terminal protein sorting signals. *Bioinformatics*, 18(2) :298–305, 2002.

[CAPPE97] J. Cedano, P. Aloy, J.A. Perez-Pons, and Querol E. Relation between amino acid composition and cellular location of proteins. *Journal Of Molecular Biology*, 266(3) :594–600, 1997.

[CC02] Kuo-Chen Chou and Yu-Dong Cai. Using Functional Domain Composition and Support Vector Machines for Pr ediction of Protein Subcellular Location. *J. Biol. Chem.*, 277(48) :45765–45769, 2002.

[Cho01] Kuo-Chen Chou. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins : Structure, Function, and Genetics*, 43 :246–255, 2001.

[CNR00] M. Cokol, R. Nair, and B. Rost. Finding nuclear localization signals. *EMBO Reports*, 1(15) :411–415, 2000.

[DBNvHB04] Jannick Dyrlov Bendtsen, Henrik Nielsen, Gunnar von Heijne, and Soren Brunak. Improved prediction of signal peptides : Signalp 3.0. *Journal of Molecular Biology*, 340 :783–795, July 2004.

[DG00] A. Drawid and M. Gerstein. A bayesian system integrating expression data with sequence patterns for localizing proteins : comprehensive application for the yeast genome. *Journal Of Molecular Biology*, 301 :1059–1075, 2000.

[EEvHC03] Olof Emanuelsson, Arne Elofsson, Gunnar von Heijne, and Susana Cristobal. In silico prediction of the peroxisomal proteome in fungi, plants and animals. *Journal of Molecular Biology*, 330 :443–456, July 2003.

[ENBvH00] Olof Emanuelsson, Henrik Nielsen, Søren Brunak, and Gunnar von Heijne. Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. *J. Mol. Biol.*, 300 :1005–1016, 2000.

[HFG+03] Won-Ki Huh, James V. Falvo, Luke C. Gerke, Adam S. Carroll, Russell W. Howson, Jonathan S. Weissman, and Erin K. O'Shea. Global analysis of protein localization in budding yeast. *Nature*, 425 :686–691, October 2003. 10.1038/nature02026.

[HL04] Ying Huang and Yanda Li. Prediction of protein subcellular locations using fuzzy k-NN method. *Bioinformatics*, 20(1) :21–28, 2004.

[HN96] P Horton and K Nakai. A probabilistic classification system for predicting the cellular localization sites of proteins. *Proc Int Conf Intell Syst Mol Biol.*, 4 :109–115, 1996.

[HS01] Sujun Hua and Zhirong Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8) :721–728, 2001.

[IIS+04] Tadashi Imanishi, Takeshi Itoh, Yutaka Suzuki, Claire O'Donovan, Satoshi Fukuchi, Kanako O. Koyanagi, Roberto A. Barrero, Takuro Tamura, Yumi Yamaguchi-Kabata, Motohiko Tanino, Kei Yura, Satoru Miyazaki, Kazuho Ikeo, Keiichi Homma, Arek Kasprzyk, Tetsuo Nishikawa, Mika Hirakawa, Jean Thierry-Mieg, Danielle Thierry-Mieg, Jennifer Ashurst, Libin Jia, Mitsuteru Nakao, Michael A. Thomas, Nicola Mulder, Youla Karavidopoulou, Lihua Jin, Sangsoo Kim, Tomohiro Yasuda, Boris Lenhard, Eric Eveno, Yoshiyuki Suzuki, Chisato Yamasaki, Jun-ichi Takeda, Craig Gough, Phillip Hilton, Yasuyuki Fujii, Hiroaki Sakai, Susumu Tanaka, Clara Amid, Matthew Bellgard, Maria de Fatima Bonaldo, Hidemasa Bono, Susan K. Bromberg, Anthony J. Brookes, Elspeth Bruford, Piero Carninci, Claude Chelala, Christine Couillault, Sandro J. de Souza, Marie-Anne Debily, Marie-Dominique Devignes, Inna Dubchak, Toshinori Endo, Anne Estreicher, Eduardo Eyras, Kaoru Fukami-Kobayashi, Gopal R. Gopinath, Esther Graudens, Yoonsoo Hahn, Michael Han, Ze-Guang Han, Kousuke Hanada, Hideki Hanaoka, Erimi Harada, Katsuyuki Hashimoto, Ursula Hinz, Momoki Hirai, Teruyoshi Hishiki, Ian Hopkinson, Sandrine Imbeaud, Hidetoshi Inoko, Alexander Kanapin, Yayoi Kaneko, Takeya Kasukawa, Janet Kelso, Paul Kersey, Reiko Kikuno, Kouichi Kimura, Bernhard Korn, Vladimir Kuryshev, Izabela Makalowska, Takashi Makino, Shuhei Mano, Regine Mariage-Samson, Jun Mashima, Hideo Matsuda, Hans-Werner Mewes, Shinsei Minoshima, Keiichi Nagai, Hideki Nagasaki, Naoki Nagata, Rajni Nigam, Osamu Ogasawara, Osamu Ohara, Masafumi Ohtsubo, Norihiro Okada, Toshihisa

Okido, Satoshi Oota, Motonori Ota, Toshio Ota, Tetsuji Otsuki, Dominique Piatier-Tonneau, Annemarie Poustka, Shuang-Xi Ren, Naruya Saitou, Katsunaga Sakai, Shigetaka Sakamoto, Ryuichi Sakate, Ingo Schupp, Florence Servant, Stephen Sherry, Rie Shiba, Nobuyoshi Shimizu, Mary Shimoyama, Andrew J. Simpson, Bento Soares, Charles Steward, Makiko Suwa, Mami Suzuki, Aiko Takahashi, Gen Tamiya, Hiroshi Tanaka, Todd Taylor, Joseph D. Terwilliger, Per Unneberg, Vamsi Veeramachaneni, Shinya Watanabe, Laurens Wilming, Norikazu Yasuda, Hyang-Sook Yoo, Marvin Stodolsky, Wojciech Makalowski, Mitiko Go, Kenta Nakai, Toshihisa Takagi, Minoru Kanehisa, Yoshiyuki Sakaki, John Quackenbush, Yasushi Okazaki, Yoshihide Hayashizaki, Winston Hide, Ranajit Chakraborty, Ken Nishikawa, Hideaki Sugawara, Yoshio Tateno, Zhu Chen, Michio Oishi, Peter Tonellato, Rolf Apweiler, Kousaku Okubo, Lukas Wagner, Stefan Wiemann, Robert L. Strausberg, Takao Isogai, Charles Auffray, Nobuo Nomura, Takashi Gojobori, and Sumio Sugano. Integrative annotation of 21,037 human genes validated by full-length cdna clones. *PLoS Biology*, 2 :e162, June 2004.

[KAH+02] Anuj Kumar, Seema Agarwal, John A. Heyman, Sandra Matson, Matthew Heidtman, Stacy Piccirillo, Lara Umansky, Amar Drawid, Ronald Jansen, Yang Liu, Kei-Hoi Cheung, Perry Miller, Mark Gerstein, G. Shirleen Roeder, and Michael Snyder. Subcellular localization of the yeast proteome. *Genes Dev.*, 16(6) :707–719, 2002.

[KCC03] Yu-Dong Cai Kuo-Chen Chou. Prediction and classification of protein subcellular location ?- ?sequence-order effect and pseudo amino acid composition. *Journal of Cellular Biochemistry*, 90 :1250–1260, 2003.

[KKS04] Lukas Käll, Anders Krogh, and Erik L. L. Sonnhammer. A combined transmembrane topology and signal peptide prediction method. *Journal of Molecular Biology*, 338 :1027–1036, May 2004.

[LSG+04] Z. Lu, D. Szafron, R. Greiner, P. Lu, D.S. Wishart, B. Poulin, J. Anvik, C. Macdonell, and R. Eisner. Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, 20(4) :547–556, 2004.

[MAPM04] Jakob Christian Mueller, Christophe Andreoli, Holger Prokisch, and Thomas Meitinger. Mechanisms for multiple intracellular localization of human mitochondrial proteins. *Mitochondrion*, 3(6) :315–325, May 2004.

[McG85] Duncan J. McGeoch. On the predictive recognition of signal peptide sequences. *Virus Research*, 3 :271–286, October 1985.

[MSBP02] Richard Mott, Jorg Schultz, Peer Bork, and Chris P. Ponting. Predicting Protein Cellular Localization Using a Domain Projection Method. *Genome Res.*, 12(8) :1168–1174, 2002.

[NBM+05] Mitsuteru Nakao, Roberto A. Barrero, Yuri Mukai, Chie Motono, Makiko Suwa, and Kenta Nakai. Large-scale analysis of human alternative protein isoforms : pattern classification and correlation with subcellular localization signals. *Nucl. Acids Res.*, 33(8) :2355–2363, 2005.

[NCR03] Rajesh Nair, Phil Carter, and Burkhard Rost. NLSdb : database of nuclear localization signals. *Nucl. Acids Res.*, 31(1) :397–399, 2003.

[NH99] Kenta Nakai and Paul Horton. Psort : a program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends in Biochemical Sciences*, 24 :34–35, January 1999.

[NK92] K. Nakai and M. Kanehisa. A knowledge base for predicting protein localization sites in eucaryotic cells. *Genomics*, 14(4) :897–911, 1992.

[NMSE+03a] Georg Neuberger, Sebastian Maurer-Stroh, Birgit Eisenhaber, Andreas Hartig, and Frank Eisenhaber. Motif refinement of the peroxisomal targeting signal 1 and evaluation of taxon-specific differences. *Journal of Molecular Biology*, 328 :567–579, May 2003.

[NMSE+03b] Georg Neuberger, Sebastian Maurer-Stroh, Birgit Eisenhaber, Andreas Hartig, and Frank Eisenhaber. Prediction of peroxisomal targeting signal 1 containing proteins from amino acid sequence. *Journal of Molecular Biology*, 328 :581–592, May 2003.

[NN92] H Nakashima and K Nishikawa. The amino acid composition is different between the cytoplasmic and extracellular sides in membrane proteins. *FEBS Letter*, 303 :141–146, 1992.

[NR02a] Rajesh Nair and Burkhard Rost. Inferring sub-cellular localization through automated lexical analysis. *Bioinformatics*, 18(90001) :78S–86, 2002.

[NR02b] Rajesh Nair and Burkhard Rost. Sequence conserved for subcellular localization. *Protein Sci*, 11(12) :2836–2847, 2002.

[NR03] Rajesh Nair and Burkhard Rost. Loc3d : annotate sub-cellular localization for protein structures. *Nucl. Acids Res.*, 31(13) :3337–3340, 2003.

[NR05] Rajesh Nair and Burkhard Rost. Mimicking cellular sorting improves prediction of subcellular localization. *Journal of Molecular Biology*, 348 :85–100, April 2005.

[PCST00] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin dags for multiclass classification. In S.A. Solla, T.K. Leen, and K.-R. Mueller, editors, *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.

[PK03] Keun-Joon Park and Minoru Kanehisa. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19(13) :1656–1663, 2003.

[RH98] A Reinhardt and T Hubbard. Using neural networks for prediction of the subcellular location of proteins. *Nucl. Acids. Res.*, 26(9) :2230–2236, 1998.

[SCLK05] Deepak Sarda, Gek Chua, Kuo-Bin Li, and Arun Krishnan. pslip : Svm based protein subcellular localization prediction using multiple physicochemical properties. *BMC Bioinformatics*, 6(1) :152, 2005.

[SLHT04] M. Scott, G. Lu, M. Hallett, and D. Y. Thomas. The Hera database and its use in the characterization of endoplasmic reticulum proteins. *Bioinformatics*, 20(6) :937–944, 2004.

[STH04] Michelle S. Scott, David Y. Thomas, and Michael T. Hallett. Predicting Subcellular Localization via Protein Motif Co-Occurrence. *Genome Res.*, 14(10a) :1957–1966, 2004.

[vH86] G. von Heijne. A new method for predicting signal sequence cleavage sites. *Nucleic Acids Research*, 14 :4683–4690, June 1986.

[WW99] J. Weston and C. Watkins. Multi-class support vector machines. In *Proceedings ESANN*, 1999.

[YDXJXbKC02] Cai Yu-Dong, Liu Xiao-Jun, Xu Xue-biao, and Chou Kuo-Chen. Support vector machines for prediction of protein subcellular location by incorporating quasi-sequence-order effect. *Journal of Cellular Biochemistry*, 84 :343–348, 2002.

[Yua99] Z Yuan. Prediction of protein subcellular locations using markov chain models. *FEBS letter*, 451 :23–26, 1999.

Chapitre 6

Méthodologie pour la classification

Les concepts et méthodes énoncés aux chapitres 2 et 3 l'ont été dans un cadre mathématique général, en se référant dès que possible à des exemples simples pour illustrer notre propos. Cependant, de par la nécessité d'un énoncé clair et précis, la pureté théorique des définitions et théorèmes peut avoir par moment camouflé une partie des difficultés propres à la mise en place d'une stratégie d'apprentissage réellement adaptée au problème posé.

Nous présentons dans ce chapitre les méthodes mises en oeuvre pour adapter une stratégie de classification au problème de prédiction du compartiment cellulaire à partir de la séquence protéique. En particulier, le problème bioinformatique présente plusieurs caractéristiques spécifiques :

- Le nombre de classes prises en compte peut être élevé. En général on doit classer une protéine parmi 5 à 12 compartiments.
- Les effectifs des différents compartiments sont extrêmement variables. Pour certains jeux de données les rapports d'effectifs observés entre deux compartiments peuvent atteindre un facteur de l'ordre de 30. Ce point est aussi dû aux biais possibles d'expérimentation, mais il n'est pas à négliger.
- Les données disponibles et annotées expérimentalement sont en nombre limité, ce qui peut nuire au pouvoir de généralisation sur les compartiments les moins peuplés.
- Le but de ces classifications est, à terme, de pouvoir fournir au biologiste une piste d'annotation pour la protéine, que ce soit lors d'un travail de fouilles sur des génomes complets, pour guider l'expertise du spécialiste, ou lors d'une utilisation occasionnelle. Il est donc souhaitable de pouvoir disposer d'un indice de qualité sur la prédiction. Aussi, l'évaluation de la précision du classificateur doit être menée avec le plus grand soin.

Nous présenterons dans un premier temps les indicateurs de classification qui seront utilisés par la suite.

Ensuite, après avoir vu de quelle manière les problèmes multiclasses peuvent être

abordés avec l'approche SVM, nous présenterons la méthode de classification que nous proposons. L'idée suivie est de ramener le problème multiclasse à une suite de classifications binaires structurées dans un arbre de décision.

L'arbre est construit par regroupements successifs sur les compartiments suivant la "ressemblance" de modèles de Chaînes de Markov cachées ajustés sur chacun. Cette approche permet ainsi d'entraîner les classificateurs aux premiers noeuds de l'arbre sur un plus grand nombre d'exemple, et ainsi d'obtenir des résultats plus fiables malgré une perte de précision sur le compartiment prédit.

Une alternative à cette procédure de contruction "ascendante" est aussi proposé, procédant par découpages successifs du groupe de compartiments. Cette seconde méthode, qui utilise des critères géométriques, permet aussi de procéder aux découpages à l'aide de distances fondées sur les noyaux présentés au chapitre précédent.

Dans la suite on considérera donc le problème de prédiction sur l'ensemble des compartiments $C = \{c_1, \ldots, c_{|C|}\}$. Chaque compartiment c_i est composé de n_i séquences (avec $\sum_{c_j \in C} n_j = n$), et on notera les séquences du compartiment $c_i = \left\{\wp_1^i, \ldots, \wp_{n_i}^i\right\} = \wp_\bullet^i$.

6.1 Evaluation de performance

Afin d'obtenir une estimation sur les performances du classificateur, la méthode la plus couramment utilisée est celle de la validation croisée (voir 3.1). Cependant, si le risque fournit un critère naturel de minimisation, il est souhaitable de pouvoir calculer d'autres indices de qualité lors de l'évaluation d'une méthode de classification. Par exemple, concernant la localisation subcellulaire des protéines, lors de l'affectation d'un compartiment à une protéine, il peut être intéressant de connaître le pourcentage de protéines qui seraient prédites à tort comme appartenant à ce compartiment.

Introduisons donc les indicateurs qui sont couramment utilisés pour comparer les performances de différents classificateurs. Commençons avec la matrice de confusion, qui résume le résultat d'une procédure de classification. Considérons une expérience de classification par validation croisée N fois. on définit la matrice de confusion M^l associée au test l comme la matrice à n lignes et n colonnes qui compte en ligne i et colonne j le nombre de protéines appartenant au compartiment c_i et prédits dans le compartiment c_j. On note M la matrice résumant l'ensemble des résultats par validation croisé : $M = \sum_{i=1}^{N} M^i$. Ainsi, le nombre total de protéines bien prédites est donné par $\sum_{i=1}^{|C|} M(i,i)$.

On définit ensuite pour chaque compartiment c_i :

$$\mathrm{TP}_i = M(i,i) \qquad \text{(le nombre de vrais positifs dans le compartiment } c_i\text{)}$$

$$\mathrm{FP}_i = \sum_{\substack{j=1 \\ j\neq i}}^{|C|} M(i,j) \qquad \text{(le nombre de faux positifs } \textit{i.e.} \text{ prédits à tort comme étant dans } c_i\text{)}$$

$$\mathrm{FN}_i = \sum_{\substack{j=1 \\ j\neq i}}^{|C|} M(j,i) \qquad \text{(le nombre de faux négatifs)}$$

$$\mathrm{TN}_i = n - \mathrm{TP}_i - \mathrm{FP}_i - \mathrm{FN}_i \qquad \text{(les vrais négatifs)}$$

$$\mathrm{sens}_i = \frac{\mathrm{TP}_i}{\mathrm{FN}_i + \mathrm{TP}_i} \qquad \text{(la sensibilité, proportion des individus de } c_i \text{ bien classés.)}$$

$$\mathrm{spec}_i = \frac{\mathrm{TN}_i}{\mathrm{TN}_i + \mathrm{FP}_i} \qquad \text{(la spécificité, ou qualité de la prédiction sur le compartiment } c_i\text{.)}$$

$$\mathrm{MCC}_i = \frac{\mathrm{TP}_i \times \mathrm{TN}_i - \mathrm{FP}_i \times \mathrm{FN}_i}{\sqrt{(\mathrm{TP}_i + \mathrm{FP}_i) \times (\mathrm{TP}_i + \mathrm{FN}_i) \times (\mathrm{TN}_i + \mathrm{FP}_i) \times (\mathrm{TN}_i + \mathrm{FN}_i)}}$$

La dernière quantité est le coefficient de corrélation de Matthews associée au compartiment c_i. Ce coefficient peut être préféré pour la robustesse de l'estimation faite sur la qualité de prédiction. Pour la prédiction du compartiment cellulaire, on regardera généralement la sensibilité et la spécificité pour chaque compartiment, qui s'interprètent plus intuitivement. Ces quantités permettent ainsi de statuer sur la proportion de protéines classées à juste titre dans chaque compartiment. Néanmoins, la comparaison entre plusieurs méthodes peut s'avérer ardue si elle porte sur une douzaine de couples. On peut ainsi regarder la sensibilité globale :

$$\mathrm{sens}_g = \frac{1}{n} \sum_{i=1}^{|C|} \mathrm{TP}_i = \frac{1}{n} \sum_{i=1}^{|C|} M(i,i)$$

Cette quantité peut biaiser favorablement les résultats dans le cas où les effectifs des classes sont très variables. En effet, prenons un exemple où deux compartiments représentent à eux seuls 75% des effectifs, et que la tâche du classificateur consiste à classer parfaitement sur ces deux compartiments, on obtiendra une sensibilité globale de 75% sur le jeu de données.

On regardera donc aussi la sensibilité moyenne :

$$\mathrm{sens}_m = \frac{1}{|C|} \sum_{i=1}^{|C|} \mathrm{sens}_i$$

Enfin, l'analyse de la courbe ROC (pour "receiver operation characteristic") peut aussi être pratique lors de la comparaison entre plusieurs classificateurs. La courbe ROC correspond au tracé de la proportion de vrais positifs contre la proportion de vrai négatifs en faisant varier le seuil choisi pour la fonction de classification. Ceci permet d'avoir une idée sur le comportement de la fonction de décision suivant la stringence appliquée à la détermination de la classe. Une classification aléatoire donnera donc une courbe ROC proche de la première bissectrice. Ainsi une mesure fréquemment utilisée est l'aire sous la courbe (ou AUC[1]), possiblement translatée de $\frac{1}{2}$, pour l'aire sous la première bissectrice.

Aussi, afin d'avoir une idée sur la stabilité des résultats, et donc de la robustesse du classificateur, on peut calculer un écart-type à partir des N matrices de confusion sur les différents indicateurs susnommés.

6.2 Classification multiclasse

Les méthodes présentées en 3.3 pour la classification par SVM portaient sur des problèmes à deux classes. Les résultats présentés sur les séparateurs à vaste marge peuvent être étendus aux cas multiclasses. Par exemple [WW99] propose de résoudre le problème d'optimisation quadratique suivant :

$$\begin{aligned} &\min_{\mathbf{w}_r,\xi^r,b_r} \frac{1}{2}\sum_{i=1}^{|C|}\|\mathbf{w}_r\|^2 + \frac{C}{m}\sum_{i=1}^{m}\sum_{r\neq y_i}\xi_i^r \\ &\text{soumis à } \langle \mathbf{w}_{y_i}, \mathbf{x}_i\rangle + b_{y_i} \geq \langle \mathbf{w}_r, \mathbf{x}_i\rangle + b_r + 2 - \xi_i^r \\ &\qquad \xi_i^r \geq 0 \end{aligned} \tag{6.1}$$

où $m \in \{1,\ldots,|C|\}$ y_i et $y_i \in \{1,\ldots,|C|\}$ donne la classe de l'individu $\mathbf{x}_i$.

Cependant, ces méthodes sont coûteuses en temps de calcul, du fait que le problème d'optimisation doit être résolu pour tous les Support Vectors en même temps. Une autre approche consiste à simplement ramener le problème de classification multiclasse à plusieurs problèmes binaires. Les méthodes fréquemment utilisées sont :

1-vs-rest : Pour chaque compartiment c_i, un SVM f_i est entraîné avec c_i pour la classe positive et $\cup_{j=1,j\neq i}^{|C|} c_i$ pour la classe négative. La classe d'un individu x est ensuite choisie suivant : $\arg\max_{c_i\in C} f_i(x)$. Cette stratégie est couramment appelée "winner takes all".

1-vs-1 : Ici, pour chaque couple de compartiments (c_i, c_j) on entraîne un classificateur $f_{(i,j)}$. Ainsi, $|C|(|C|-1)/2$ SVM sont entraînés. La classe est ensuite choisie par vote des $f_{(i,j)}$. On appelle souvent cette stratégie "max-wins".

[1] pour Area Under the Curve

Dans la pratique, malgré le fait qu'aucune borne sur le risque ne permette de justifier ces deux dernières méthodes, elles réalisent en général des performances comparables à celle du SVM multiclasse (6.1). Notons que [PCST00] ont proposé une adaptation de la stratégie de 1-vs-1, en structurant les étapes de classification dans un graphe acyclique dirigé. Ils obtiennent alors une borne "à la Vapnik" qui permet de justifier l'approche réalisée. Néanmoins, nous utiliserons dans la suite les deux méthodes "classiques", lors de l'étude des résultats.

6.3 Classification par arbre de décision

Un des problèmes que nous avions mis en avant pour la prédiction du compartiment cellulaire était la faible population de certains compartiments, et la grande variabilité des effectifs. Cependant, avant d'être adressée à sa destination finale, une protéine circule généralement à travers d'autres compartiments. Par exemple, les protéines adressées dans le noyau sont toujours synthétisées dans le cytoplasme. On pourrait donc supposer que les protéines suivant la même voie d'adressage partagent des caractéristiques communes, telles les protéines de la voie sécrétoire qui possèdent en général un peptide signal.

On propose donc d'intégrer cet a priori biologique par la construction d'un arbre correspondant à des regroupements successifs sur les compartiments. Une séquence est ensuite prédite par classifications successives à chaque noeud de l'arbre. L'élaboration de l'arbre pourrait être guidée par un a priori biologique, par exemple en "mimant" les processus d'adressage connus pour les protéines (ce cas sera aussi traité par la suite). Nous proposons ici deux méthodes qui construisent l'arbre en supposant qu'un modèle CMC génère les séquences conditionnellement à chaque compartiment.

On notera $\mathcal{M}_i$ la CMC estimée sur le compartiment c_i. $c_{i\cup j}$ désignera le regroupement des compartiments c_i et c_j et $\mathcal{M}_{i\cup j}$ le modèle associé estimé sur les séquences $\wp_\bullet^{i\cup j}$.

6.3.1 Construction de l'arbre

Notons qu'afin d'être en mesure de construire l'arbre par regroupements successifs, il est nécessaire de disposer d'une distance calculable entre tous les couples de compartiments et modèles ajustés.

Idéalement, on voudrait pouvoir statuer sur la similarité entre deux compartiments sur la base de deux critères simultanés :

- la proximité des modèles dans l'espace des paramètres, éventuellement doté d'une métrique tenant compte de la variance des estimateurs.
- l'évolution de l'ajustement des individus, des deux modèles uniques au modèle correspondant à leur fusion.

Bien entendu, on ne peut pas privilégier tous ces critères simultanément, mais nous proposons ici deux distances couramment utilisées et portant l'accent sur des points différents :

1. La distance fondée sur le rapport de vraisemblance :

$$d_{\mathrm{ML}}(c_i, c_j) = d_{\mathrm{ML}}(i, j) = -\log\left(\frac{\prod_{k=1}^{n_i+n_j} \mathbb{P}(x_k^{i\cup j}|\mathcal{M}_{i\cup j})}{\prod_{k=1}^{n_i} \mathbb{P}(x_k^i|\mathcal{M}_i)\ \prod_{k=1}^{n_j} \mathbb{P}(x_k^j|\mathcal{M}_j)}\right)$$

   Notons que cette quantité tend à augmenter linéairement avec la taille des séquences. Dans notre cas, où le rapport d'effectifs entre différents compartiments peut être de l'ordre de 20 ou 30, ces effets de "taille" peuvent alors favoriser certains regroupements. Cependant, cette distance permet aussi de prendre en compte conjointement deux paramètres différents : la proximité des lois de probabilités associées aux modèles et la variance des exemples par rapport au modèle estimé.

2. La distance en variation totale entre les CMC estimées $\mathcal{M}_i$ et $\mathcal{M}_j$ (voir la définition 2.9). Cette distance se calcule après réécriture de la CMC dans l'alphabet $\Sigma \times \mathcal{S}$ et on la note $d_{VT}(i, j)$ pour $d_{VT}(\mathcal{M}_i, \mathcal{M}_j)$. Rappelons (section 2.2.2) que les modèles CMC ne sont identifiables que sur l'espace quotient des permutations d'états. Ainsi lors du calcul de la distance en variation totale entre deux modèles $\mathcal{M}_i$ et $\mathcal{M}_j$, on testera toutes les permutations possibles sur l'étiquetage des états. On choisira comme distance $d_{VT}(\mathcal{M}_i, \mathcal{M}_j)$ celle réalisant le minimum sur les $|\mathcal{S}|!$ possibilités.

Remarquons quelques autres propriétés sur ces deux distances. La distance en variation totale ne prend en compte ni les tailles des échantillons, ni la variabilité des estimateurs.

D'un autre côté, la distance en vraisemblance est adaptée au test classique de rapport de vraisemblance, qui dit que :

$$2\, d_{ML}(i, j) \sim \chi^2(|\boldsymbol{\theta}_i| + |\boldsymbol{\theta}_j| - |\boldsymbol{\theta}_{i\cup j}|)$$

Où $\boldsymbol{\theta}_i$,$\boldsymbol{\theta}_j$, $\boldsymbol{\theta}_{i\cup j}$ sont respectivement les paramètres de $\mathcal{M}_i$, $\mathcal{M}_j$ et $\mathcal{M}_{i\cup j}$, les modèles ayant été estimés par maximum de vraisemblance. Dans tous les premiers tests, les significativités obtenues pour le rassemblement de compartiments étaient au plus de l'ordre de 10^{-15}, et n'amenaient donc jamais à rejeter un regroupement. Ceci s'explique par la taille des ensembles d'exemples, qui génèrent alors des rapports de vraisemblance de forte valeur.

Dans la pratique, il arrive que EM ne converge que vers un maximum local de la vraisemblance dans le cas de petits compartiments. Alors, pour certaines fusions où EM converge "mieux", les distances par maximum de vraisemblance deviennent négatives, faussant l'algorithme de construction de l'arbre. Par la suite, on a donc préféré utiliser la distance en variation totale .

6.3.1.1 Méthode “Bottom-up”

L’algorithme de regroupement présenté ci dessous (algorithm 2) diffère peu des classification ascendantes hiérarchiques (CAH) visant à construire l’arbre de saut minimal. On procède récursivement sur le jeu de noeuds parents en décrémentant l’ensemble C par fusionnement des deux compartiments de distance minimale. Remarquons cependant que dans le cas des CAH, la matrice des distances après regroupement est déduite de la précédente par l’utilisation d’un critère d’agrégation tel que le “single linkage” ou critère de Ward. Dans notre cas, une réestimation complète du modèle (et des distances correspondantes) a lieu sur les compartiments fusionnés après chaque regroupement.

Algorithm 2 Algorithme de construction de l’arbre

Requis : Ensemble de compartiments C,
Ensemble des noeuds parents $\mathcal{N}$ ($\sim C$ à l’initialisation)
$\forall c_i \in C$, estimer la CMC $\mathcal{M}_i$
Tant que $|C| > 1$ **Faire**
Pour tout $c_i, c_j \in C, i > j$ **Faire**
calculer $d(i, j) = d_{\mathrm{ML}}(c_i, c_j)$ ou $\mathrm{d_{VT}}(\mathcal{M}_i, \mathcal{M}_j)$
Fin Pour
$(i_0, j_0) \leftarrow argmin(d(i, j))$
retirer c_{i_0} et c_{j_0} de C
ajouter $c_{i_0 \cup j_0}$ à C
fusionner les noeuds n_{i_0} et n_{j_0} dans $\mathcal{N}$
estimer $\mathcal{M}_{i_0 \cup j_0}$
Fin Tant que

Notons que cette méthode ne construit pas nécessairement l’arbre possédant la meilleure sensibilité globale. Au contraire, les compartiments sont agrégés en partant des feuilles, et la classification procède à partir de la racine. De plus, tout comme les méthodes de classification ascendante hiérarchique classiques (type single linkage), aucun critère d’agrégation sur la variance n’est pris en compte. Les effets de peigne connus dans le cas des CAH, pourraient dans notre cas être amplifiés par les grandes différences d’effectifs entre chaque compartiment.

6.3.1.2 Méthode “Top-down”

Du fait du nombre relativement faible de compartiments, on peut aussi imaginer une méthode alternative de construction de l’arbre, débutant cette fois de la racine. Tester dans ce cas tous les partitionnements en deux groupes possibles à chaque noeud par estimation de CMC est impraticable. Nous proposons donc une méthode grossière de division à chaque noeud, fondée sur le score de Fisher de chacune des séquences.

Supposons disposer de l'ensemble de compartiments C_n au noeud n, et de la CMC $\mathcal{M}_n$ associée. La division suivante est déterminée ainsi :

1. $\forall \wp_j^i \in c_j$, et $c_j \in C_n$, on calcule le score de Fisher de $\wp_j^i$: $\overrightarrow{\mathcal{V}_{\mathcal{F}}}(\wp_j^i, \mathcal{M}_n)$
2. $\forall c_j \in C_n$, on calcule la variance sur les scores de Fisher :

$$\mathcal{V}_{c_j}^{\mathcal{M}_n} = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} \left\| \mathbf{g}_i^{\mathcal{M}_n} - \overrightarrow{\mathcal{V}_{\mathcal{F}}}(\wp_j^i, \mathcal{M}_n) \right\|^2 \quad \text{et la variance globale} \quad \mathcal{V}_{C_n}^{\mathcal{M}_n}$$

3. On prend la partition à deux éléments sur C_n qui minimise la somme des variances intra classe, ou maximise la variance interclasse (ce qui est équivalent d'après la relation de Huyghens).

Cette méthode demande au plus $(|C| - 1)$ évaluations de modèles (le nombre de noeuds de l'arbre) et reste donc raisonnable en temps de calcul.

Bien entendu, on pourrait reformuler ces découpages descendants comme un problème d'estimation où on chercherait à estimer un mélange de deux CMC à chaque noeud, en interdisant au mélange d'avoir lieu au milieu d'un compartiment.

En regardant l'algorithme de découpage proposé ci dessus, on voit qu'en fait on travaille implicitement dans l'espace d'attributs induit par le noyau de Fisher (cf 3.3.4.2). En effet, pour un noyau k défini positif, on sait, d'après Mercer, qu'il existe une fonction Φ telle que $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$. Ainsi, sous certaines conditions sur le noyau k, on pourrait déduire de la matrice de Gram des individus une matrice de distance. Cet algorithme pourrait donc être appliqué avec des noyaux pour séquences de caractères, la classification se faisant ensuite à chaque noeud par un SVM.

6.3.2 Classification dans l'arbre

La figure 6.3.2 montre un arbre construit à partir du jeu de données sur l'homme et utilisant la distance en variation totale comme critère de regroupement (méthode "bottom-up"). Les distances obtenues pour les regroupements sont données comme étiquettes pour les noeuds . Ainsi, pour chaque noeud de l'arbre, le jeu de séquences déterminant la CMC associée est constituée des feuilles filles de ce noeud. Par exemple dans la figure 6.3.2, la CMC correspondant au noeud portant le label 0.18 est estimée sur les séquences du lysosome, du réticulum endoplasmique et du golgi.

Introduisons un système de numérotation des noeuds et feuilles de l'arbre :

- Chaque feuille ou noeud d'un arbre est identifiée de manière unique à l'aide d'un nombre binaire.
- la racine de l'arbre est identifiée par le nombre `0`.
- Si un noeud est identifié par le nombre η, ses enfants à gauche et à droite sont identifiés respectivement par les nombres $\eta.0$ et $\eta.1$.

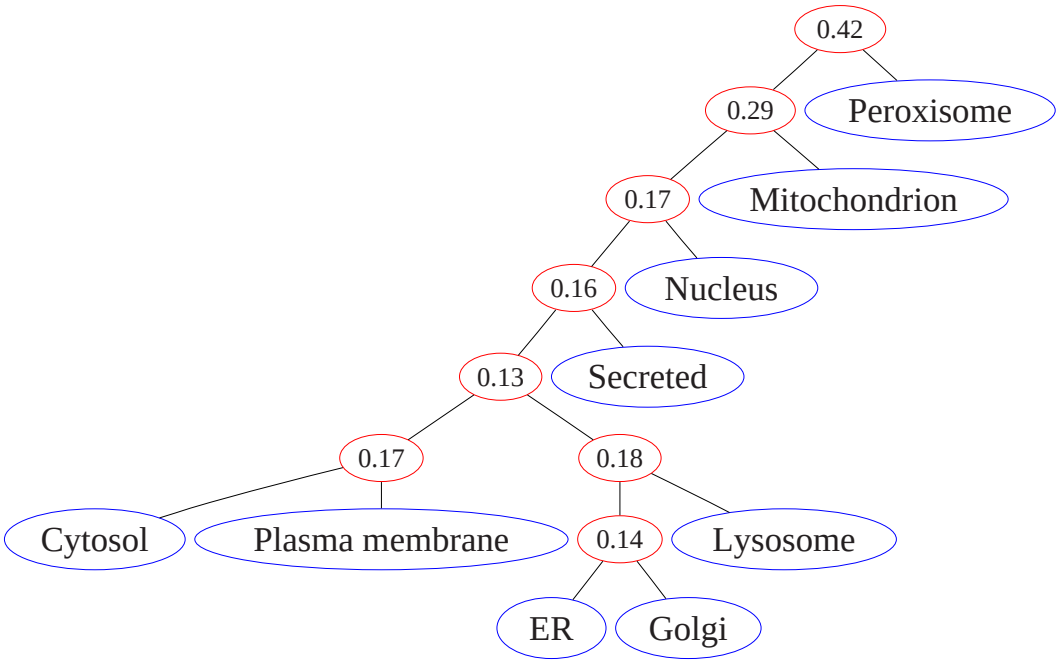


FIG. 6.1 – arbre de regroupement calculé sur un jeu de données humaines, et utilisant la distance en variation totale comme critère de regroupement. Les valeurs indiquées à chaque noeud sont les distances en variation totale obtenues pour le regroupement.

Notons & l'opérateur unaire pour le décalage d'un bit à droite. Pour un noeud η, $\&\eta$ fait référence à son père, $\&^2\eta$ à son grand père et, si le noeud est de profondeur d, $\&^d\eta$ désigne la racine.

La classification d'une séquence se décompose alors en classifications successives à chaque noeud de l'arbre, le compartiment étant attribué dès que le noeud prédit est une feuille. Bien entendu, ce principe de classification ne dépend plus du classificateur utilisé. Cependant, dans notre cas, nous sommes en mesure à chaque noeud, de calculer la probabilité que la séquence appartienne au noeud de gauche ou de droite. Nous montrons rapidement dans la section suivante comment s'écrivent les conditionnements pour permettre le calcul de la probabilité d'appartenance d'une séquence à chaque noeud.

En appliquant la loi de Bayes, la probabilité d'un noeud n de profondeur i se calcule

donc comme :

$$\mathbf{P}(\eta|X) \propto \mathbf{P}(\eta, X) \tag{6.2}$$

$$\begin{aligned}\mathbf{P}(\eta, X) &= \mathbf{P}(\eta, X, \&\eta, \&^2\eta, \ldots, \&^i\eta) \quad \text{(l'arbre est acyclique)} \\ &= \mathbf{P}(\eta|X, \&\eta, \&^2\eta, \ldots, \&^i\eta) \cdot \mathbf{P}(\&\eta|X) \qquad (6.3) \\ &\propto \mathbf{P}(\eta_i|X, \&\eta, \&^2\eta_{i-2}, \ldots, \&^i\eta) \cdot \mathbf{P}(\eta|X, \&\eta, \ldots, \&^i\eta) \ldots \mathbf{P}(\&^{i-1}\eta|X, \&^i\eta) \cdot \mathbf{P}(X|\&^i\eta) \qquad (6.4)\end{aligned}$$

L'équation 6.3 montre qu'on peut calculer les probabilités d'appartenance aux noeuds par un parcours récursif de l'arbre, n'impliquant donc pas un coût de calcul supplémentaire par rapport à la méthode initiale.

Remarquons qu'obtenir une probabilité d'appartenance à chaque compartiment permet de tenir compte de deux contraintes importantes liées à la prédiction de l'adressage. Premièrement, de par sa nature d'outil d'aide à l'annotation, la qualité des prédictions du classificateur (en d'autres termes sa spécificité) doit pouvoir être imposée lors de la prédiction, et sera dans la pratique préférée à une couverture complète des protéines (*i.e.* une prédiction pour chaque protéine). Ce point s'ajuste facilement en seuillant les probabilités d'appartenance à chaque noeud. En outre, cet indice de qualité permet aussi de fixer un niveau de précision sur le compartiment prédit. Ensuite, bien que les protéines multiclasses n'aient pas été prises en compte initialement, que ce soit dans l'ensemble d'entraînement ou par la modélisation de l'arbre, ce schéma de classification peut permettre ce type de prédictions. Nous n'avons pas utilisé cette dernière propriété, les protéines "multiclasses" ayant été éliminées du jeu de données (chapitre suivant).

Chapitre 7

Résultats

7.1 Fréquences de codons et localisation cellulaire.

Ce chapitre présente un travail mené originellement sur la levure et qui visait à déterminer si l'intégration de descripteurs extraits de la séquence nucléique permettaient une meilleure discrimination entre certains compartiments. Cette étude a porté en premier lieu sur un jeu construit en fusionnant les données Gene Ontology disponibles sur *SGD* et des données GFP mises à disposition par [HFG+03]. Ces résultats, originellement acceptés pour une présentation orale courte à JOBIM 2004 sont donc présentés dans la section 7.1.1.

Néanmoins, ces données issues d'expériences à grande échelle sont peu fiables. nous présentons donc en section 7.1.2 les vérifications ultérieures qui ont été menées sur un jeu construit à partir des annotations SGD en retirant les protéines homologues. Les résultats obtenus confirment les premières constatations concernant l'influence de l'usage du code sur certains groupes de protéines, permettant une meilleure discrimination des protéines cytoplasmiques.

On a ensuite vérifié si ce type d'information pouvait aussi s'utiliser chez des organismes supérieurs, tels que la souris et l'homme.

7.1.1 Premiers résultats

Introduction

Ce travail propose l'étude de l'influence de descripteurs globaux pour la prédiction de la localisation subcellulaire d'une protéine. Nous avons travaillé sur la levure, car les données disponibles pour cet organisme nous ont permis de regarder chaque séquence au niveau nucléotidique. En effet, certaines approches ont déjà démontré l'utilisation de codons synonymes particuliers favorisés pour certaines structures secondaires de protéines [OS98]. Nous avons examiné si cette information pouvait améliorer la qualité

de la prédiction pour certains compartiments. L'utilisation des protéines de levure a aussi été motivée par les données récemment mises à disposition par le laboratoire de O'Shea [HFG+03], nous permettant de travailler avec 1940 protéines ne présentant pas plus de 70% de similarité de séquence entre elles.

Après une études des corrélations possibles entre la localisation d'une protéine et son profil en acides aminés ou en codons, nous avons procédé à une classification par Support Vector Machines (SVM) avec les fréquences des acides aminés ou des codons. Nous proposons aussi une approche alternative en classant une protéine suivant sa vraisemblance pour des modèles de markov caché (CMC), estimés à partir des séquences nucléotidiques de chacun des compartiments.

Matériel et Méthodes

Jeu de données Pour la construction du jeu de données, nous avons travaillé sur deux types de données : (1) les données annotées à partir d'expériences dans la Saccharomyces Genome Database (SGD) après mise en correspondance avec Gene Ontology (le jeu `SGD_GO`), et (2) les localisations obtenues par marquage à la protéine de fluorescence (le jeu `GFP`) proposées par le laboratoire O'Shea [HFG+03]. Comme remarqué dans l'article, les jeux `GFP` et `SGD_GO` se correspondent sur plus de 80%. Afin de disposer de suffisamment de données tout en préservant la qualité des annotations, nous avons procédé au regroupement par plusieurs étapes : (1) réannotation des localisations du jeu `GFP` à l'aide des termes de Gene Ontology, (2) intersection des annotations entre les jeux `GFP` et `SGD_GO`, (3) rajout des entrées de chacun des deux jeux n'étant pas présentes dans l'autre, (4) retrait des protéines présentant une trop forte similarité de séquence. Les effectifs pour chaque compartiment après les étapes (3) et (4) sont donnés dans la table 7.1. Remarquons par ailleurs que dans le cas du jeu de données `GFP`, 815 protéines sont annotées comme cytoplasmiques *et* nucléaires. Nous avons choisi de retirer ces protéines de l'analyse.

Comme la totalité de la séquence est utilisée pour la classification, nous avons regroupé les protéines en clusters, suivant leur similarité de séquence. Ceci permettant de prévenir une surestimation des performances de classification. Ainsi deux protéines sont considérées comme appartenant au même cluster si elles possèdent plus de 40% de similarité en pleine longueur (matrice BLOSUM62). Tous les alignements deux à deux ont été réalisés avec le programme `FASTA` [PL88]. La table 7.2 résume les annotations compartimentales des protéines appartenant aux 6 plus grands regroupements obtenus. Contrairement à ce qu'on pouvait penser au premier abord, ces clusters possèdent une composition compartimentale hétérogène. Finalement, le jeu de données utilisé pour l'analyse a été construit en prenant une protéine au hasard dans chaque cluster. Les localisations cellulaires correspondant au "cell wall", à la périphérie du noyau, à l'endosome et au péroxisome ont aussi été retirées du jeu de données en raison d'effectifs trop faibles.

compartiment GFP	fusion GFP et SGD_GO	sans redondance
cell periphery	112	49 ⋆
cellwall [1]	37	6
cytoplasm [2]	859	528 ⋆
cytoplasm,nucleus	815	538
cytoskeleton [3]	135	91 ⋆
endosome [4]	35	28
ER [5]	300	213 ⋆
golgi [6]	99	68 ⋆
mitochondria [7]	551	404 ⋆
nuclear periphery	54	34
nucleolus [8]	70	50 ⋆
nucleolus,nucleus	104	72
nucleus [9]	649	463 ⋆
peroxisome [10]	49	27
vacuolar membrane [11]	54	31
vacuole [12]	127	75 ⋆
total	4050	2677

TAB. 7.1 – Effectifs du jeu de données créé après regroupement des jeux GFP et SGD_GO (colonne 2), puis après réduction des effectifs sur les protéines possédant une trop grande similarité de séquence (colonne 3). Les lignes marquées par une étoile sont celles qui seront utilisées pour l'analyse. Les identifieurs Gene Ontology utilisés pour la correspondance avec le jeu de données GFP sont les suivants : [1]GO :5618, [2]GO :5829, [3]GO :5856, [4]GO :5768, [5]GO :5783, [6]GO :5794, [7]GO :5758 GO :5759 GO :5740, [8]GO :5730, [9]GO :5694 GO :5634,[10] GO :5777, [11]GO :9705, [12]GO :5773.

Support Vector Machines L'étape de classification sur les fréquences en acides aminés et codons a été menée avec une approche par Support Vector Machine (SVM [Vap95, Vap98]). Partant de vecteurs descriptifs $\mathbf{x}_i \in \mathbb{R}^d$ étiquetés sur deux classes -1 et $+1$ (ici, chaque $\mathbf{x}_i$ correspond aux fréquences de tous les acides aminés et/ou les codons), la méthodologie par SVM construit une fonction séparatrice en replongeant les données dans un espace d'attributs de grande dimension et en trouvant l'hyperplan maximisant la marge entre les deux classes. L'intérêt des SVM est aussi qu'il n'est pas nécessaire de travailler explicitement dans cet espace d'attributs, les individus étant comparés deux à deux par leur produit scalaire. La fonction noyau réalise alors cette projection. Dans la suite, nous avons utilisé un noyau RBF qui calcule la similarité entre deux points $\mathbf{x}_i$ et $\mathbf{x}_j$ à l'aide de la formule suivante (où g est un paramètre du noyau correspondant à la

	c1	c2	c3	c4	c5	c6
cell periphery	19	7	6			4
cellwall	4	24				
cytoplasm	64	27	8		2	6
cytoplasm,nucleus	64	20	10		12	1
cytoskeleton	17	3	1			
endosome		1	1			
ER	4	8	1	1		1
golgi	4		3	26		
mitochondria	6	7				3
nuclear periphery	3	9				
nucleolus	4	8				
nucleolus,nucleus	4	18				
nucleus	54	13	1		9	
peroxisome	1	1		1		2
vacuolar membrane	4					3
vacuole	6	17	1			1
total	258	163	31	28	23	21

TAB. 7.2 – répartition compartimentale pour les 6 plus grands clusters obtenus en regardant la similarité de séquence.

largeur du noyau) :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{g}\right)$$

Ensuite, si on considère n individus, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \{-1; +1\}$, la fonction séparatrive peut se déduire en résolvant le problème d'optimisation quadratique sous contrainte suivant :

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i$$

$$\text{soumis à :} \quad \sum_{i=1}^{n} y_i\alpha_i = 0$$

$$\forall i = 1, \ldots, n, \quad 0 \leq \alpha_i \leq \frac{C}{n}$$

où C est le terme contrôlant le compromis choisi entre l'erreur de classification et la taille de la marge. Un vecteur à tester $\mathbf{t}$ est ensuite classé suivant le signe de $\sum_i y_i\alpha_i K(\mathbf{x}_i, \mathbf{t}) + b$.

Comme ici, le problème de classification posé est multiclasse, nous avons appliqué une stratégie de classification en 1 contre 1. Ainsi, tous les classificateurs à deux groupes sont entraînés (soit 36 classificateurs pour 9 compartiments). Une protéine testée est ensuite assignée au compartiment pour lequel elle possède le nombre maximum de votes. Dans le cas où plusieurs classes possèderaient le même nombre de votes, nous avons tiré au hasard l'un des compartiments en prenant en compte les proportions relatives dans chacune des classes estimées sur l'ensemble d'entraînement. Les calculs ont été menés à l'aide du logiciel libsvm [CL01]. Les paramètres *c* (qui ajuste le compromis entre erreur et taille de la marge) et *g* ont été ajustés en contrôlant la sensibilité moyenne sur l'ensemble des compartiments.

Chaînes de Markov cachées Reprenant le travail de [Yua99] qui utilisait des modèles markoviens pour la classification, nous avons affiné sa méthode en incorporant des modèles de Markov caché (CMC) estimés sur la séquence nucléotidique. Cette modélisation nous a permis de : (1) prendre en compte les contraintes de phase sur les séquences codantes et (2) rendre possible la succession de régions de composition différente le long de la séquence d'un gène.

Finalement, nous avons proposé pour chaque compartiments trois modèles différents, en proposant respectivement, 2, 3 et 4 états cachés pour les séquences codantes (modèles notés respectivement 2_coding, 3_coding et 4_coding). L'ordre des chaînes de Markov associées à chaque état caché a été fixé à 2, permettant ainsi de prendre en compte la composition en trinucléotide de la séquence. Une séquence de test est alors classée suivant sa vraisemblance par rapport aux modèles estimés sur chacun des compartiments. Tous les calculs ont été menés à l'aide du programme SHOW (Structured HOmogeneities Watcher [NBM+02]) qui avait déjà été utilisé avec succès pour la détection de gènes sur des organismes procaryotes.

Resultats

Analyse discriminante sur les fréquences en acides aminés et codons. Afin de pouvoir évaluer les corrélations possibles entre les fréquences en acides aminés ou codons et la localisation dans un compartiment spécifique, nous avons dans un premier temps mené une analyse linéaire discriminante (LDA). La figure 7.1 résume les résultats obtenus par la LDA sur la composition en acides aminés. Les trois premiers axes participent à 88% pour la variance du nuage (axe 1 : 58%,axe 2 : 20%, et axe 3 : 10%).

Sur le premier axe, un groupe de protéines du réticulum endoplasmique est clairement séparé du reste du groupe (points verts sur la droite). Ce groupe consiste en 42% de toutes les protéines du réticulum endoplasmique. Le premier axe est majoritairement construit par les acides aminés polaires et chargés, corrélés positivement, contre les deux plus gros résidus hydrophobiques sur la partie négative.

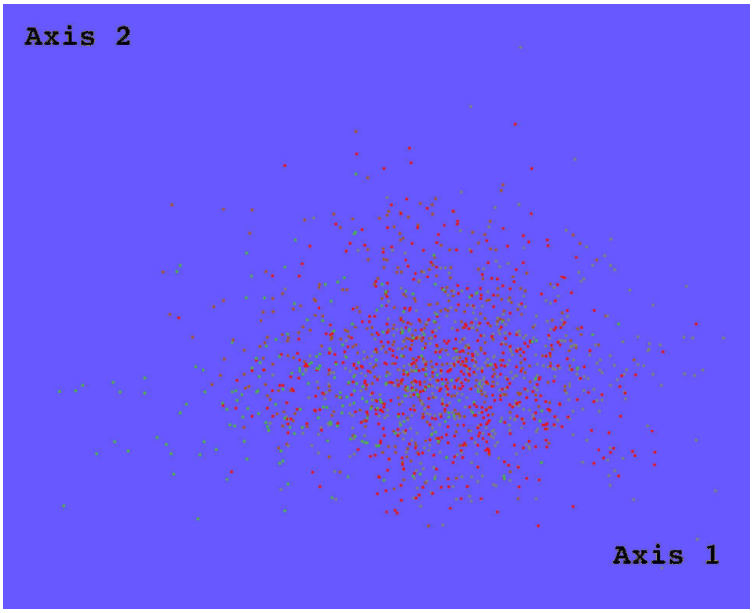


FIG. 7.1 – Représentation des protéines des 4 compartiments les plus peuplés pour les deux premiers axes obtenus par l'analyse discriminante sur les fréquences en acides aminés. Chaque point du graphique correspond ainsi à une protéine (codes couleurs : rouge=cytoplasm, vert =ER, marron = mitochondria and gris=nucleus)

Le second axe sépare une partie des protéines cytoplasmiques et mitochondriales (45 cytoplasmiques, 44 mitochondriales et 22 nucléaires) du reste des protéines. Alors que cet axe est construit par les résidus R,K,P,H,W,M et L pour la partie positive, et S,V pour la partie négative, 34,5% des protéines de ce groupe sont connues comme appartenant à la sous unité ribosomale. Remarquons que pour ce jeu de données, 135 protéines cytoplasmiques et mitochondriales appartiennent à la sous-unité du ribosome, correspondant à 14,5% de la population totale. Enfin, le troisième axe est moins caractérisé, car il sépare un petit groupe de protéines cytoplasmiques.

L'axe est composé de CH pour la partie positive et LMWRQN pour la partie négative.

Pour évaluer si les fréquences de codons pouvaient comporter une information supplémentaire, nous avons ensuite procédé à une LDA en utilisant les fréquences globales des codons comme descripteurs (figure 7.2).

Dans ce cas, les trois premiers axes contribuent à 81% de la variance globale (axe 1 : 42%, axe 2 : 26% et axe 3 : 13%).

Ici, le premier axe sépare aussi une partie des protéines du réticulum endoplasmique. En outre, les codons contribuant à la construction de cet axe codent les acides aminés ayant construit le premier axe pour la LDA sur les fréquences en AA. Une propriété intéressante apparaît sur le second axe. En effet, cet axe sépare mieux que les acides

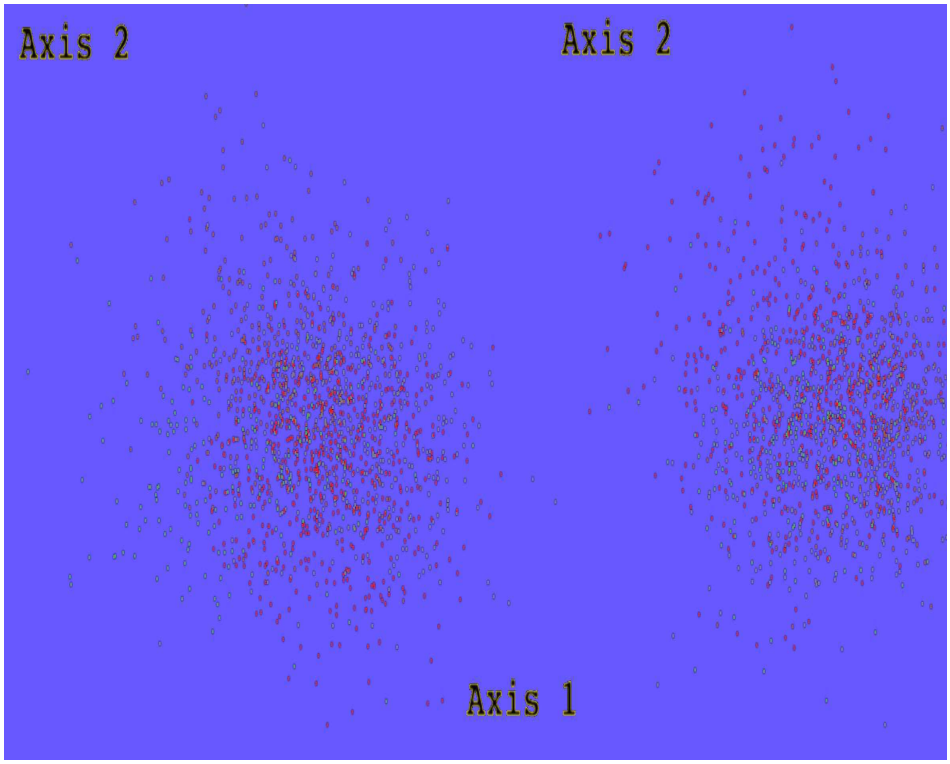


FIG. 7.2 – Représentation des protéines des 4 compartiments les plus peuplés pour les trois premiers axes obtenus par l'analyse discriminante sur les fréquences en codons. Les codes couleur sont les même que pour la figure 7.1

aminés un groupe de protéines mitochondriales pour les valeurs positives, ainsi qu'un groupe de protéines cytoplasmiques pour les valeurs négatives. Cette séparation est dûe aux deux codons les plus rares de R et P, favorisés dans le groupe de protéines mitochondriales, et pour des résidus hydrophobiques et polaires, favorisés dans le groupe de protéines cytoplasmiques.

Le troisième axe est aussi surprenant, en ce qu'un groupe de protéines cytoplasmiques est séparé sur la partie supérieure. Sur ces 89 protéines cytoplasmiques, 46 correspondent à des composants structurels de la sous-unité ribosomale (68 possèdent cette annotation sur la totalités des protéines cytoplasmiques). La séparation sur cet axe est réalisée par les résidus C,H,G,R et A contre S,L,T,A,F,V et I.

Classification Les mesures de performance ont toujours été réalisées par validation croisée cinq fois. Dans le cas de la classification par SVM, chaque validation croisée a été faite dix fois afin de pouvoir estimer une déviation standard sur les sensibilités calculées pour chaque compartiment.

Contrairement à ce que les résultats de l'analyse discriminante semblaient suppo-

ser, l'utilisation des fréquences de codons comme descripteurs ne permettent par une amélioration des sensibilités sur le cytoplasme, la mitochondrie ou le noyau – les trois compartiments semblant le mieux séparés par la LDA. Afin d'évaluer si ce fait n'était pas dû avant tout à un trop grand nombre de compartiments, nous avons procédé à des classifications sur les quatre plus gros compartiments : le cytoplasme, le noyau, le réticulum endoplasmique et la mitochondrie. Les résultats, présentés dans la table 7.3, montrent que l'utilisation des fréquences de codon améliore la sensibilité globale de 11% et la sensibilité moyenne de 2,8% (la sensibilité sur le cytoplasme augmente de 8%, et de 4% sur le noyau et la mitochondrie). Malgré une mauvaise sensibilité sur le cytoplasme en utilisant les CMC, (une chute de 30% à 15% en utilisant le modèle 2_coding), on peut remarquer que le modèle 3_coding possède la meilleure sensibilité globale.

	SVM		HMM		
	AA	Codons	2_coding	3_coding	4_coding
cytoplasm	56,3 (0,9)	63,3 (1,4)	32	43	47
ER	50,6 (1,4)	46,7 (1,2)	63	64	61
mitochondria	65,5 (1,4)	69,8 (1,0)	63	76	75
nucleus	59,7 (1,1)	63,6 (1,5)	67	70	65
Global accuraccy	58,8 (0,4)	62,8 (0,6)	54,0	61,6	61,1
Mean accuracy by location	58,0 (0,4)	60,8 (0,6)	56,3	63,1	62,1

TAB. 7.3 – Resultats sur les quatre compartiments les plus peuplés, les paramètres utilisés pour les SVM sont c=8, g=32 pour les AA and c=2,g=32 pour les codons. Dans le cas de la classification par SVM, la deviation standard sur dix validations croisées est donnée entre parenthèses.

La table 7.4 présente les résultats obtenus par classification sur les fréquences des AA et des codons en utilisant des SVM, ainsi que ceux obtenus par les CMC sur 9 compartiments. On peut remarquer que les CMC possèdent dans tous les cas une meilleure sensibilité moyenne, en raison d'une meilleure sensibilité sur les compartiments les moins peuplés. Cependant, la sensibilité globale reste moindre en raison principalement d'une mauvaise sensibilité sur le cytoplasme. On peut aussi constater une chute de sensibilité pour certains compartiments tels que le nucléole ou la vacuole, quand le nombre d'états cachés augmente. Ceci peut être expliqué par un trop grand nombre de paramètres à estimer aux vues des effectifs du compartiment.

Discussion

L'utilisation de l'information provenant de la séquence nucléotidique a été proposé afin d'améliorer les performances de classification pour la localisation cellulaire sur les séquences de la levure. [HL04] a publié des résultats sur un jeu de protéines ne

	SVM				HMM		
					num. of hidden states		
	AA		Codons		2	3	4
	sens. (stdev)		sens. (stdev)		sens.(%)		
cell periphery	8,4	(3)	17,1	(1,7)	27	25	25
cytoplasm	55,6	(1,1)	52,5	(2,9)	27	36	41
cytoskeleton	8,5	(2,4)	15	(2,9)	35	39	36
ER	46,9	(2,1)	43,4	(1,9)	42	43	43
golgi	1,0	(1)	7,1	(2,6)	16	18	16
mitochondria	64,4	(0,9)	62,4	(1,8)	55	69	68
nucleolus	3,2	(1,4)	8,2	(2,2)	25	15	6,3
nucleus	58,3	(1,7)	47,4	(3,1)	39	49	51
vacuole	3,4	(1,4)	9,2	(2,2)	32	24	17
Global accuracy	48,4	(0,8)	46,8	(0,9)	37,5	44,8	45,9
Mean accuracy by location	27,7	(0,6)	29,8	(0,7)	33,2	35,2	33,8

TAB. 7.4 – Résumé de la classification en utilisant les SVM et la vraisemblance pour 9 compartiments. La sensibilité est donnée en pourcentage sur chaque compartiment. Dans le cas de la classification par SVM, la déviation standard obtenue sur dix validations croisées est donnée entre parenthèses. Les paramètres utilisés pour les SVM sont : c=16 g=32 pour les AA et c=32000 g=64 pour les codons. Pour l'approche par CMC, l'ordre fixé sur les observations est de 2.

présentant pas plus de 50% d'identité de séquences, ce qui reste comparable au seuil fixé pour la création de notre jeu de données. Bien que certains compartiments restent mal classés, les CMC montrent une amélioration pour la mitochondrie (69% contre 36,6%), le réticulum endoplasmique (43% contre 11,1%), le cytosquelette (39% contre 28,6%) et la vacuole (32% contre 6,9%). Les résultats sont encourageants pour la classification à l'aide de CMC. Nous prévoyons d'intégrer d'autres informations biologiques dans la structure en états cachés (en modélisant par exemple les signaux d'adressage) afin d'améliorer les performances de classification.

L'examen de la composition compartimentale des clusters de forte similarité de séquence soulève un problème particulier. En effet, il apparaît que des protéines possédant une forte similarité de séquence appartiennent à des compartiments différents. Ce problème mériterait une analyse plus poussée, afin de comprendre dans quelle mesure la similarité entre deux protéines peut être corrélée à leur localisation cellulaire.

Le groupe de protéines annotées comme étant cytoplasmique et nucléaire soulèvent aussi un problème intéressant. Comme certaines de ces protéines cyclent entre le cytoplasme et le noyau, il devient difficile de statuer dans quel compartiment celles ci devraient être classées. Ceci nous amène à la question plus générale concernant la prise

en compte de “compartiments fonctionnels” qui pourraient être créés en fonction de certains processus de la cellule, tels que, par exemple, les protéines cyclant entre le cytoplasme et le noyau.

7.1.2 Vérifications

Les résultats présentés à la section précédente nécessitent cependant plusieurs vérifications. En effet, on voudrait dans un premier temps vérifier si le gain en sensibilité obtenu sur les protéines nucléaires et cytoplasmiques ne serait pas dû à un biais attribuable au jeu de données GFP. On s’attachera donc à vérifier si ce résultat est confirmé sur les séquences de la levure extraites uniquement de SGD. Comme présenté en section 4.2.2, le jeu de séquences a été réduit de manière à ce que deux protéines ne présentent pas plus de 40 % de similarité de séquence. On examinera ainsi dans quelle mesure cette amélioration de la performance peut être expliquée par le biais dans l’usage du code chez la levure.

Rappelons rapidement l’état des connaissances actuelles sur le biais d’usage des codons. Dans chaque génome, tous les codons synonymes pour un même acide aminé ne sont pas utilisés uniformément. Pour les micro-organismes à croissance rapide, tels que *E. Coli* ou *S. Cerevisiae*, on a pu établir que les codons utilisés préférentiellement sont corrélés positivement avec le taux d’ARN de transfert correspondant, dans la cellule [Ike85]. Ainsi le choix de certains codons pour un ARN messager influera sur l’efficacité de sa traduction. Pour ces deux organismes (et dans une moindre mesure pour *C. Elegans* et *D. Melanogaster*), on a pu montrer que cette relation peut être expliquée par une pression de sélection sur l’efficacité de la traduction, ce qui n’est pas le cas chez les mammifères.

Une explication possible pourrait être que les proportions relatives des ARN de transferts isoaccepteurs sont différents suivant le type cellulaire. En guise de contrôle négatif, on a donc simplement examiné si l’utilisation de fréquences de codons pouvait influer sur les performances de classification pour des jeux de séquences construites sur l’Homme et la Souris.

Le tableau 7.5 reporte les résultats de sensibilité obtenus sur les données SGD pour une classification par SVM en stratégie 1vs1 pour la classification, donc identique à celle présentée en 7.1.1. Les paramètres du noyaux ont été déterminé en prenant comme critère d’optimisation la sensibilité moyenne.

Comme on peut le voir, en comparant aux données `GFP_GO` et `SGD` (`GFP_SGD`) de la table 7.3, le gain en sensibilité globale est de plus de 10 % en utilisant les fréquences des acides aminés (58,8% contre 69,1%), et de 12% en utilisant les fréquences des codons (62,8% contre 74,8%). Ceci peut être expliqué essentiellement par la forte augmentation de la sensibilité (> 30%) dans le cas du noyau (53,7% contre 85,8% pour avec les AA et 63,6% contre 91,7% en utilisant les fréquences des codons). En effet, le jeu construit à partir de SGD contient 651 protéines nucléaires, contre 463 pour le jeu

compartiment	AA sensibilité (en %)	Codons
cytoplasme	41,0 (1,6)	**59.8** (1,0)
Réticulum Endoplasmique	45,3 (1,7)	46,1 (2,0)
Mitochondrie	55,8 (1,4)	57,2 (1,0)
Noyau	85,8 (0,7)	**91,7** (0,5)
sens. globale	69,1	74,8
sens. moyenne	57,0	63,7

TAB. 7.5 – résultats de classification sur les quatre compartiments les plus peuplés. Les paramètres utilisés pour les SVM sont $c = 2$, $g = 64$ pour les AA et $c = 2048$, $g = 0.0625$ pour les codons. Les valeurs entre parenthèses correspondent aux écarts types estimés en répétant cent fois une validation croisée dix fois.

GFP_SGD. Ces 188 protéines de différence avaient été annotées comme cytoplasmiques ou cytoplasmiques et nucléaires dans le jeu GFP et donc retirées pour l'apprentissage.

Le phénomène inverse peut être constaté concernant la mitochondrie, où la sensibilité chute de plus de 10%. En effet, 289 protéines sont identifiées comme mitochondriales dans le jeu de données non redondantes SGD, contre 404 pour GFP_SGD et la concordance est de plus de 95 % sur les séquences SGD et GFP.

En examinant maintenant la table 7.5, on constate que l'utilisation des fréquences de codons améliore la sensibilité globale de manière significative et en particulier sur le cytoplasme (+ 18%) et le noyau (+ 6%). Ces observations renforcent donc celle qui avaient été faite sur le tableau 7.3.

Intéressons nous plus en détail au gain de performance observé sur les protéines cytoplasmique. En entrainant un SVM sur le cytoplasme contre les 3 autres compartiments regroupés, on obtient alors une sensibilité de 43,9% (40% de spécificité) en utilisant les fréquences des acides aminés et de 55% (89% de spécificité) avec les fréquences des codons. Nous avons identifié les protéines impliquées dans cette amélioration en procédant à une validation croisée 5 fois (les résultats restaient stables sur plusieurs répétitions aléatoires). Nous les détaillons à la suite.

Parmi les protéines du cytosol, l'utilisation des fréquences de codons permet la classification de 29 protéines prédites comme cytoplasmiques par les fréquences des acides aminés. Sur ces 29 protéines, 23 sont annotées dans SGD comme composants structurels du ribosome. Ces protéines sont connues comme étant traduites à une grande vitesse, et possèdent donc un biais d'usage des codons favorisant les codons fréquents. En effet, le CAI (Codon Adaptation Index) moyen [SL87] de ces 29 gènes est de 0.74, ce qui est d'ailleurs peu étonnant, sachant que cet indice a été mis au point majoritairement à l'aide des protéines ribosomales.

Regardons maintenant les protéines expliquant la bien meilleure spécificité sur le cytoplasme. Toujours pour la même expérience, 83 protéines (35 mitochondriales, 32

nucléaires et 16 du réticulum endoplasmique) sont prédites comme cytoplasmiques par le SVM sur les acides aminés en étant correctement écartés par le descripteur sur les codons. De manière intéressante, le CAI moyen des gènes correspondant est de 0,19 (écart type de 0.08). En effet, parmi les 32 protéines nucléaires des 83 prédites comme cytoplasmiques, 12 correspondent à des facteurs de transcription. De même, parmi les 35 protéines mitochondriales, 10 sont des oxydoréductases et 9 des transporteurs.

Comme présenté en section 5.4, pour le développement de leur système expert, [DG00] intégraient des données de puce à ADN dans leur classifieur bayésien pour prédire la localisation cellulaire d'une protéine chez la levure. [DJG02] ont ensuite étudié le lien pouvant exister entre le niveau d'expression moyen d'un gène et la localisation cellulaire de son produit. Les auteurs remarquent que les protéines localisées dans le cytoplasme présentent un niveau d'expression moyen très élevé. De même les niveaux d'expression moyen pour les protéines mitochondriales et nucléaires sont très bas. De plus, en croisant ces données d'expression avec les annotation fonctionnelles de la base MIPS[1], il apparaît que les ARN messagers possédant un fort niveau d'expression codent entre autre des protéines ribosomales, liées à la synthèse de protéines ou à la glycolise. Ceux avec un faible niveau d'expression seraient impliquées dans le transport, la transcription, et la synthèse des ARN de transfert.

Pour la Souris et l'Homme, la même méthode de validation croisée dix fois a été utilisée. Cette étape de validation croisée a ensuite été répétée 100 fois afin de calculer des écarts types sur les sensibilités obtenues. Comme on peut le constater dans le tableau 7.6, aucune différence significative ne peut être constatée entre l'utilisation d'acides aminés ou de codons comme descripteurs pour prédire la localisation cellulaire de la protéine dans un des quatre compartiments regardés.

Une validation sur une bactérie pourrait être intéressante puisque l'on sait que chez ces organismes, les protéines extracellulaires sont plus fortement exprimées.

paramètres	Souris				Homme			
	AA		Codons		AA		Codons	
	$c = 8, g = 64$		$c = 32, g = 16$		$c = 8, g = 64$		$c = 32, g = 16$	
cytoplasme	50,5	(1,8)	53,0	(1,9)	50,5	(1,8)	53,0	(1,9)
Extracellulaire	65,0	(2,2)	66,7	(2,4)	65,0	(2,2)	66,7	(2,4)
Mitochondrie	42,3	(2,8)	38,6	(2,6)	42,3	(2,8)	38,6	(2,6)
Noyau	67,9	(1,5)	69,5	(2,0)	67,9	(1,5)	69,5	(2,0)
sens. globale	59,0		60,1		59,0		60,1	
sens. moyenne	56,4		56,9		56,4		56,9	

Tab. 7.6 – Comparaison des sensibilités sur les jeu de souris et d'homme pour quatre compartiments

[1] `http ://mips.gsf.de/`

7.2 Résultats de classification

Ce chapitre fait suite aux méthodes décrites en section 7.1.1 proposant la prédiction du compartiment cellulaire à l'aide de Chaînes de Markov Cachées (CMC). En effet, dans un schéma de classification à quatre compartiments (tableau 7.3), les résultats sont comparables à ceux obtenus à l'aide de SVM sur les fréquences. Si ces CMC étaient estimées sur la séquence nucléotidique, celles présentées par la suite seront toujours à valeurs sur l'alphabet des acides aminés. En effet, les premiers tests sur la levure n'avaient pas montré de différence notable des performances, et la séquence nucléotidique ne semblerait pas apporter d'informations pertinentes dans le cas des jeux de données sur des protéines humaines et a fortiori dérivées de SWISSPROT.

Si la méthode LOCtree (section 5.2.1) prend en compte la composition moyenne en acides aminés sur les 50 premiers résidus, on verra à la section suivante comment l'estimation de CMC sur chaque compartiment, sans a priori biologique, permet de faire ressortir une information relative aux signaux d'adressage en leur associant des états cachés. En effet, sur le jeux de données déduit de Swissprot, chaque groupe de signal d'adressage N-terminal est associé à un état caché.

Ensuite, après une présentation des arbres de classification construits à l'aide de la méthode détaillée à la section 6.3, on examinera le gain qu'on peut obtenir avec cette méthode en comparaison d'une simple classification bayésienne. Bien que les arbres construits ne puissent pas réellement être justifiés biologiquement, cette méthode permet une augmentation de la sensibilité globale allant de 15% à 20%. Chaque résultat sera aussi comparé à ceux publiés sur le même jeu de données.

Enfin, pour permettre des comparaisons avec les méthodes proposant une classification sur 4 ou 5 compartiments, des arbres de construction "artisanale" et reproduisant l'a priori biologique sur l'adressage ont été construits. Bien que les résultats ne soient pas du niveau de la meilleure méthode actuelle, ils restent comparables aux méthodes plus anciennes telles que SubLoc ou NNPSL et motivent de multiples améliorations.

Les CMC utilisées dans ce chapitre ont été toujours spécifiées comme comportant trois états cachés. Ce nombre a été choisi parce qu'il donnait les meilleurs résultats lors des premières expériences sur la levure (tableaux 7.3 et 7.4). Les matrices de transitions conditionnelles à l'état caché ont été prises avec un ordre fixé à 0 ou 1.

7.2.1 Caractéristiques des CMC ajustées

La figure 7.3 représente le graphe des états cachés pour un modèle ajusté sur les séquences du réticulum endoplasmique du jeu de Park & Kanehisa. Pour plus de lisibilité, les probabilités de transitions entre états cachés inférieures à 0.001 n'ont pas été tracées. Remarquons les deux états 'begin' et 'end' qui sont associés respectivement au début et à la fin de la séquence. Ces deux états n'émettent pas d'observation et sont

introduits pour permettre de travailler conditionnellement à la longueur des séquences. Notons aussi que toutes les séquences sont supposées indépendantes pour l'estimation.

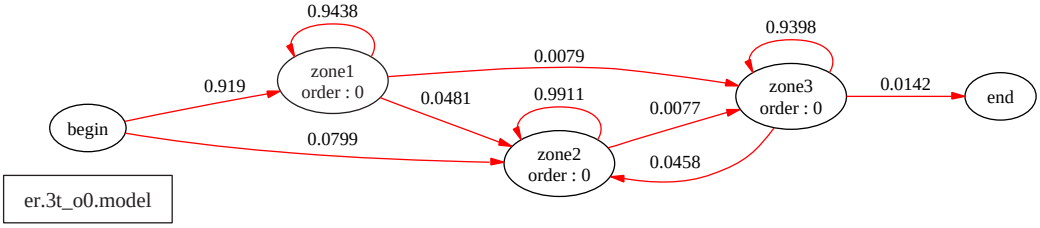


FIG. 7.3 – Graphe des états cachés pour une CMC estimée sur les protéines du réticulum endoplasmique pour le jeu de Park (ordre 0 sur les observations).

La CMC estimée possède certaines caractéristiques bien particulières. Premièrement, les débuts de séquence sont majoritairement associés à l'état zone1, très rarement atteint par la suite. Cette propriété pourrait en effet corroborer la présence connue du peptide signal aux extrémités N-terminales de ces protéines.

En outre, sur cette même figure, l'état zone3 est le seul possédant une transition vers l'état 'end'. Nous verrons aussi dans la section suivante que la composition de cette état le met en relation avec les signaux de type `KDEL` de rétention dans le réticulum endoplasmique.

Ainsi, par la suite, nous nous attacherons à regarder pour les modèles ajustés sur chaque compartiment :

- si certains états cachés sont préférentiellement associés aux débuts ou fins de séquences.
- les caractéristiques compositionnelles des états identifiés.
- le lien avec les connaissances biologiques en matière d'adressage.

Les initialisations de l'algorithme EM utilisées, sont à chaque fois similaires, avec plusieurs initialisations aléatoires pour les matrices d'observations. Les probabilités de transition entre états sont fixées à des valeurs égales (soit $\alpha(u,v) = \frac{1}{3} \forall\, u, v \in \mathcal{S}$).

Les signaux d'adressages sont associés à des états cachés Regardons pour chaque compartiment la plus grande probabilité de transition partant de l'état symbolisant le début de séquence. Sur les 12 compartiments étudiés, 9 présentent une préférence nette ($\geqslant 75\%$) pour un état caché en particulier, dont 7 avec une probabilité supérieure à 90 %. Le tableau 7.2.1 résume les valeurs de ces probabilités de transition, et les temps de séjour moyens en acides aminés associés sur les états correspondant.

Ecartons directement le cas du péroxisome, où l'état caché correspond alors à un acide aminé, causé par une surparamétrisation du modèle. En effet, l'ordre sur les acides aminés ne permet pas de contraster un troisième état de meilleure vraisemblance que la méthionine de début de séquence. Cependant, en permettant un ordre 1 sur les matrices

compartiment	P	temps de séjour moyen (en A.A.)
mitochondrie	0.99	22
chloroplaste	0.80	40
rét. endoplasmique	0.92	18
extracellulaire	1	17
lysosome	0.95	21
vacuole	0.91	13
golgi	0.94	185
cytosquelette	0.72	370
péroxisome	1	1

TAB. 7.7 – résumé des caractéristiques des états majoritairement associés aux débuts de séquences dans les CMC estimées sur chacun des compartiments : plus grande probabilité de transition partant de l'état 'begin' (colonne P) et temps moyen de séjour associé pour l'état atteint.

de transition, un état reste associé aux débuts de séquence, et possède un temps de séjour moyen de 10 bases. Malheureusement, aucun signal d'ordre biologique ne peut être mis en relation avec cet état, la majorité des protéines annotées comme adressées au péroxisome possédant un signal d'adressage se trouvant sur l'extrémité C-terminale. On peut par contre remarquer que cet état est majoritairement associé à des acides aminés hydrophobes (A,G,V,L) et à la Sérine.

Les états associés aux débuts de séquence pour le golgi et le cytosquelette présente aussi une certaine particularité. En effet, si le temps de séjour moyen sur l'état de début de séquence possède une valeur variant entre 20 et 40 acides aminés pour la plupart des compartiments, le golgi et le cytosquelette présentent respectivement des valeurs de 185 et 370 acides aminés. Ce fait peut être expliqué par les distributions des longueurs de séquences observées sur ces deux compartiments. En effet, celles-ci sont nettement bimodales, avec un premier mode à 250 lettres pour les protéines golgiennes et à 600 pour celles du cytosquelette. Cette configuration sur les transitions entre états cachés permet ainsi de reproduire le caractère bimodale des lois de longueur des CMC.

Les protéines des cinq compartiments restants ont comme point commun de posséder généralement un signal d'adressage sur leur extrémité N-terminale, responsable de leur localisation. L'étude des distances en variation totale entre les matrices de transitions associées à ces états majoritaires en début de séquences font ressortir deux groupes : l'un associé à la mitochondrie et au chloroplaste (groupe (**a**)), l'autre correspondant aux autres compartiments faisant partie de la voie secrétoire (groupe (**b**)).

Bien que le modèle CMC utilisé soit de loin plus simple que les méthodes de détection présentées en 5.1, examinons si ces états de début de séquence ont bien été associés

à tout ou partie du signal d'adressage. La figure 7.4 représente la probabilité a posteriori moyenne (sur les séquences de chacun des 5 compartiments) par position pour l'état caché majoritaire en début de séquence (trait épais).

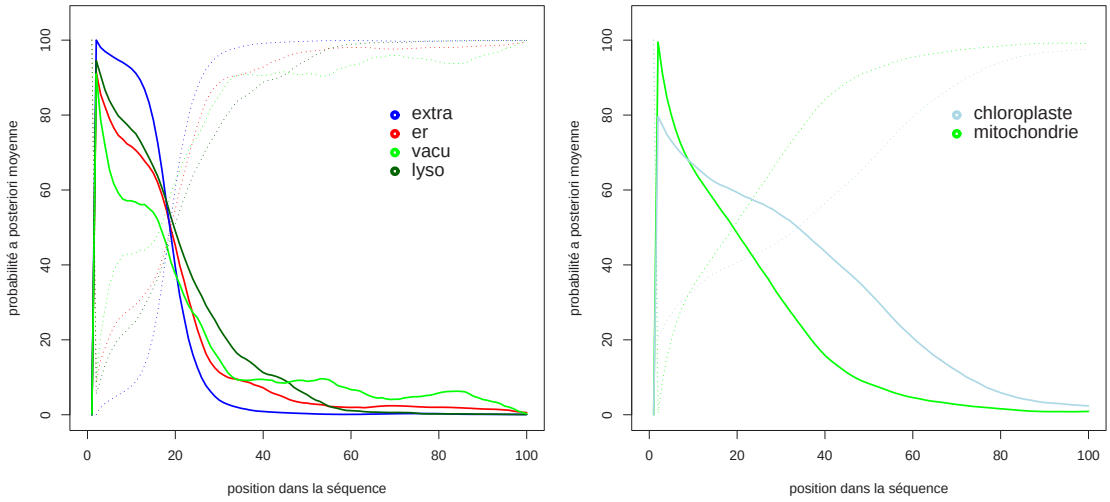


FIG. 7.4 – Probabilités moyennes par position des états cachés des groupes (**a**) et (**b**) en début de séquence. Trait épais : état caché majoritaire, pointillés : somme sur les deux états restants.

Comme on peut le constater, pour le groupe (**b**) l'état de début de séquence couvre clairement les 20 premières lettres. De même, pour le groupe (**a**), l'observation du tracé fait ressortir les longueurs moyennes données par le modèle et correspond aux connaissances sur la longueur du signal d'adressage avant le site de clivage.

La figure 7.5 présente les profils des matrices de transition estimées sur les états majoritaires en début de séquence du chloroplaste et de la mitochondrie. Leurs caractéristiques compositionnelles correspondent aux connaissances biologiques sur le sujet. En effet, les résidus acides tendent à être évités, et l'Alanine et la Sérine sont favorisés.

Comme validation supplémentaire, on a comparé ces profils aux profils déduits des jeux de données ayant servi à l'entraînement des programmes chloroP et mitoP version 1.1[2] ([ENBvH00, NEBvH97]). Pour l'estimation des fréquences, seule la séquence avant le site de clivage a été conservée. Comme on peut le voir, les deux profils correspondant au chloroplaste sont très similaires. Dans le cas de la mitochondrie, certaines différences sont remarquables, particulièrement concernant l'Alanine (A), et les acides aspartique et glutamique (D et E).

Revenons au groupe (**b**), formé par la matrice extracellulaire (extra), le réticulum endoplasmique (er), la vacuole (vacu) et le lysosome (lyso). La figure 7.6 montre les

[2] `http ://www.cbs.dtu.dk/services/TargetP/datasets/datasets.php`

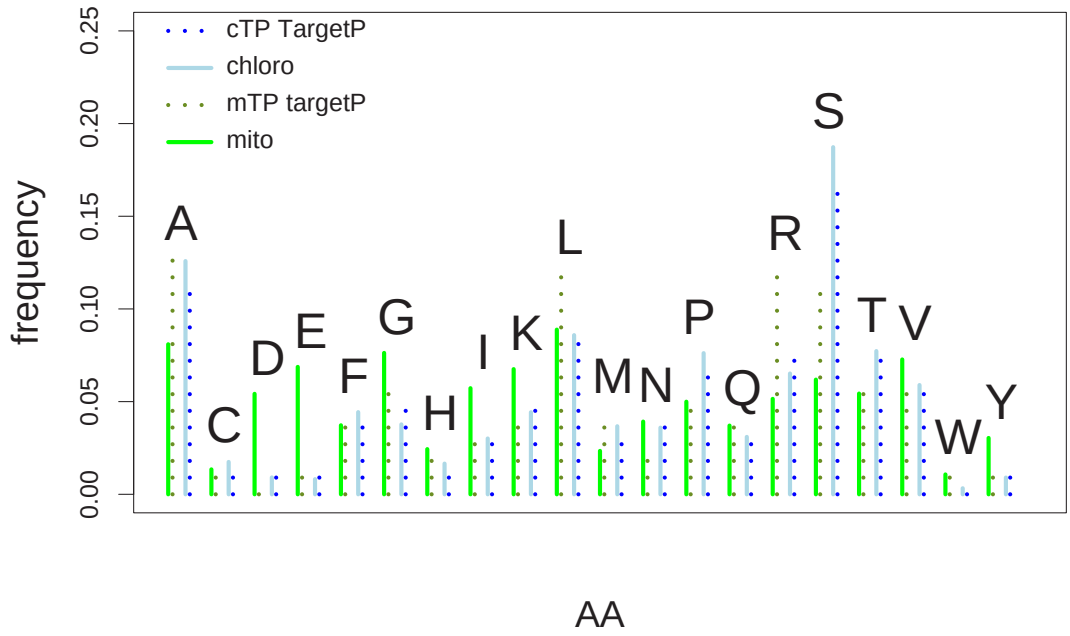


FIG. 7.5 – Profils des états cachés associés fréquemment aux débuts de séquence pour la mitochondrie et le chloroplaste. A titre de comparaison, on a rajouté en pointillé les fréquences d'acides aminés estimées à partir des jeux de données utilisés pour la construction des programmes targetP et mitoP (en se restreignant à la séquence avant le site de clivage annoté).

profils mis côte à côte pour les différents compartiments. On remarque une surreprésentation des acides aminés hydrophobes, suivis des acides aminés polaires. Le peptide signal est connu pour être composé de trois régions, la *n-région*, positivement chargée, la *h-région*, hydrophobe, puis la *c-region* polaire, suivie du site de clivage. Cet état de début de séquence s'ajuste donc vraisemblablement sur tout ou partie des trois régions. En outre, les temps de séjours associés aux états sont bien en accord avec ce qui est connu dans la littérature (longueur moyenne de 25 résidus). Normalement, ce signal devrait aussi être associé aux protéines de la membrane plasmique (plas). Si la matrice de transition sur les états cachés ne favorise pas particulièrement un état en début de séquence, ce modèle "membranaire" présente un état possèdant un profil très similaire à ceux des compartiments susnommés. Comme pour la figure 7.5, on a superposé les fréquences pour les résidus estimées à partir du jeu de données ayant servi à la constitution de signalP version 1.1. Ici aussi, les différents profils sont en accord avec les données expérimentales.

En outre, pour ce groupe de protéines, l'examen des répartitions moyennes pour les probabilités a posteriori de chaque état fait resortir un second état couramment associé à la suite de ce "simili" signal peptide. A nouveau, l'examen des matrices de transtion

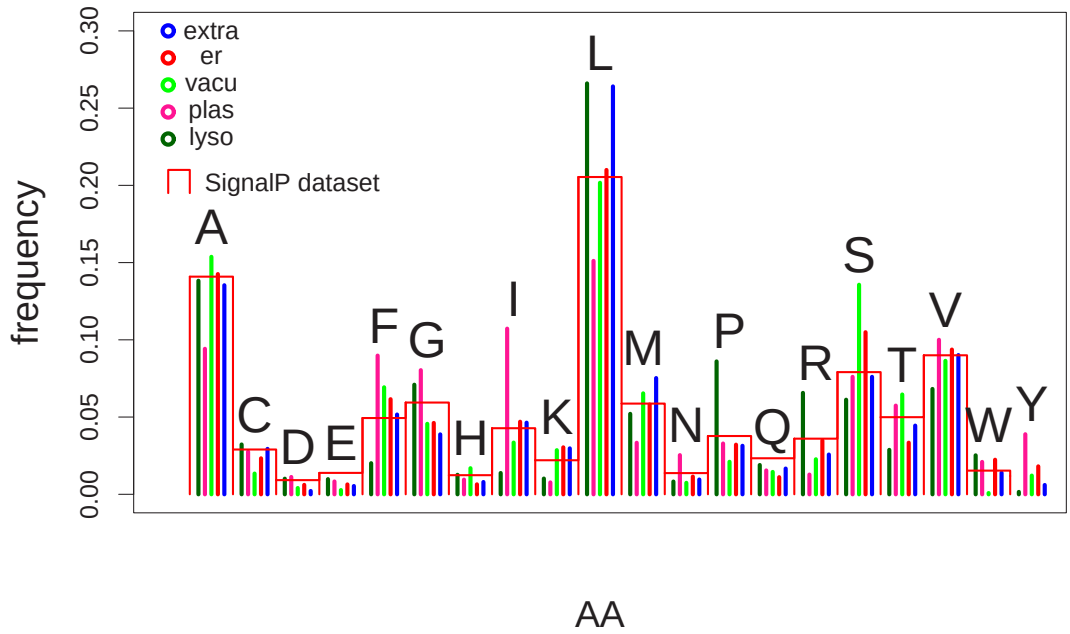


FIG. 7.6 – profil des états cachés associés fréquemment aux débuts de séquence pour le groupe des protéines empruntant la voie secrétoire. Les histogrammes annotés “SignalP” sont déduits du jeu de données ayant servi à l’entraînement du programme correspondant.

pour ces états montre qu’ils sont très proches en terme de distance en variation totale.

Essayons maintenant de statuer si, comme le laissait présager l’exemple du réticulum endoplasmique, certains états pourraient être particulièrement favorisés aux extrémités C-terminales des séquences. Pour ce faire, nous avons examiné la répartition moyenne des probabilités a posteriori au sein des séquence et pour chacun des états, sur les 100 dernières bases de chaque séquence.

Un compartiment présente des profils a posteriori, caractéristiques du choix d’un état, le réticulum endoplasmique et la membrane plasmique. Dans le cas du réticulum, le profil de l’état associé est présenté en figure (*ref*), et est compatible avec le profil prosite[3] ’`[KRHQSA]-[DENQ]-E-L<`’. En effet, les deux premières lettres du motif variant suivant les espèces, le biais de création dû à Swissprot pourrait expliquer que l’Alanine et la Lysine aient été favorisées dans le profil de la CMC.

L’étude des CMC estimées sur les jeux de données sur l’Homme et la Levure ne permet pas les mêmes généralisations. En effet, seul le signal d’adressage à la mitochondrie est reconnu dans les deux cas par le modèle estimé. Remarquons que lors de la constitutions du jeu de données de l’homme [SLHT04] remarquaient que seulement 70% des protéines annotées comme appartenant au réticulum endoplasmique (golgi

[3] motif PS00014 sur `http ://www.expasy.org/`

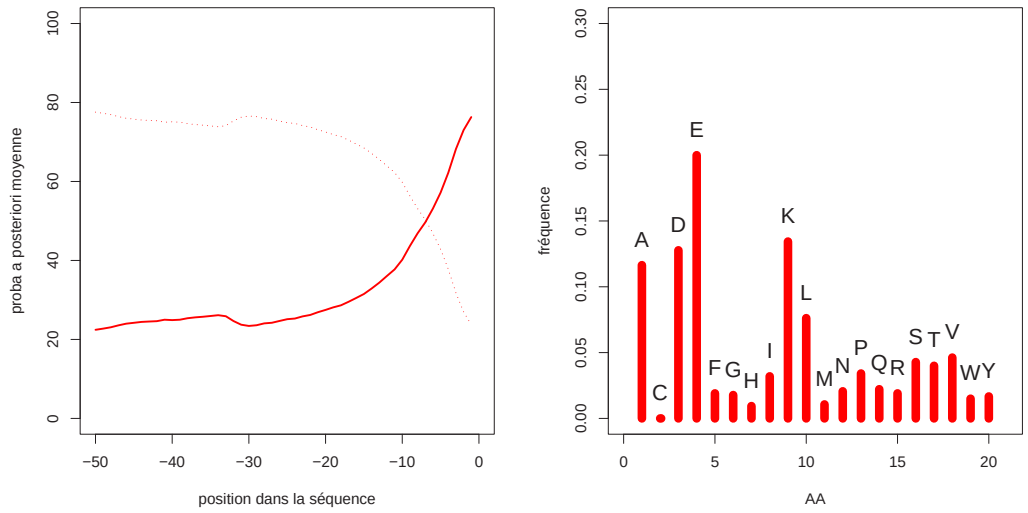


FIG. 7.7 – gauche : probabilités moyennes par position des états cachés pour la fin de séquence sur le réticulum endoplasmique, droite : profil de l'état correspondant.

30%, membrane plasmique et lysosome $\leqslant$ 80%) contenaient un peptide signal prédit par SignalP version 2.0. En outre, pas plus d'un tiers des protéines étudiées ne possédaient une version, même dégénérée d'un signal de rétention dans le réticulum. Ce fait pourrait expliquer pouquoi la CMC ne s'ajuste pas sur les séquences signal. Celà permet aussi de s'interroger sur les limitations possibles lorsqu'un jeu de données est dérivé de SWISSPROT.

7.2.2 Comparatifs des performances de classification.

Pour chaque espèce, on a présenté dans la suite les résultats obtenus avec deux des méthodes présentées précédemment : la méthode classique par comparaison des vraisemblances, et la méthode utilisant l'arbre de regroupement pour procéder aux classifications. Quand c'était possible, les résultats de classification ont aussi été comparés avec ceux mis à disposition dans les articles correspondants.

Cependant, les arbres obtenus par l'algorithme ne correspondent pas en général à la connaissance biologique sur l'adressage. On a donc ensuite proposé des classifications partant d'arbres mimant dans une certaine mesure le processus d'adressage d'une protéine.

Les arbres ont été construits à l'aide de la méthode "bottom-up" en utilisant la distance en variation totale pour chaque regroupement. En effet, comme on l'avait remarqué en section 6.3.1, si la distance du maximum de vraisemblance est utilisée comme

critère de regroupement, des valeurs de distances négatives peuvent apparaître pour certaines fusions. Ceci est attribuable à la convergence de EM vers des maxima locaux lors de certaines estimations. L'augmentation du nombre d'initialisations aléatoires de l'algorithme EM n'a pas permis de résoudre ce problème. Aussi, afin de limiter les problèmes pouvant résulter d'un surajustement des paramètres lors de l'estimation, des pseudo comptages ont été rajoutés lors les phases E de EM.

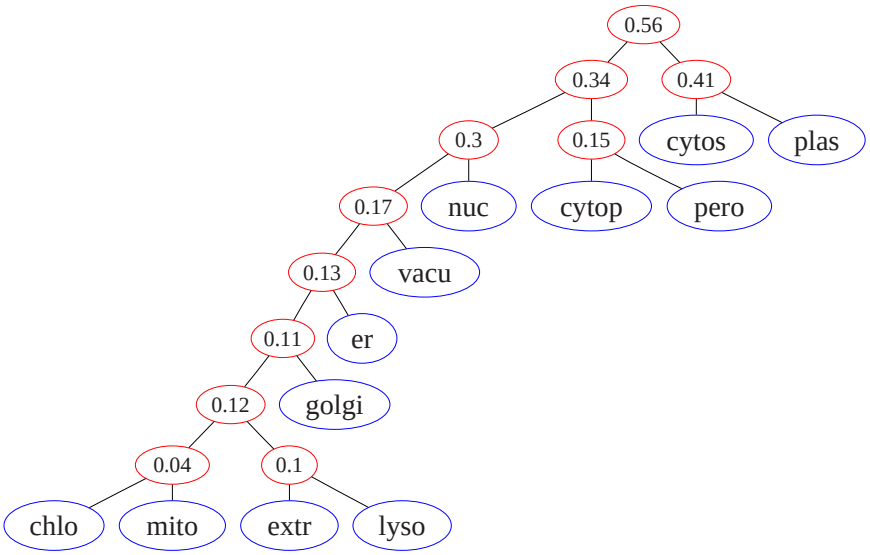


FIG. 7.8 – Arbre de regroupement construit sur le jeu de Park & Kanehisa et utilisant la distance en variation totale comme critère de regroupement (celle-ci est indiquée à chaque noeud).

La figure 7.8 montre un arbre construit à partir du jeu de données déduit de Swiss-prot [PK03] avec un ordre 0 sur les observations. Le premier regroupement, qui fusionne les protéines du chloroplaste et celles de la mitochondrie correspond bien aux connaissances sur le sujet. De même, deux compartiments faisant partie de la voie sécrétoire, la matrice extracellulaire et le lysosome sont ensuite regroupés. L'interprétation biologique des regroupements suivants n'est cependant pas aussi aisée. Il s'agit majoritairement d'adjonctions au même groupe grossissant ainsi à chaque fusion. On peut aussi remarquer que les distances de regroupement obtenues sur les noeuds sont très proches. En effet, si l'algorithme de regroupement est relancé plusieurs fois, l'arbre construit diffère fréquemment dès le second ou troisième regroupement. Ainsi, les distances en variation totale entre les modèles sont généralement trop faibles en regard de la variance liée à l'estimation par l'algorithme EM. Ce phénomène est aussi constaté sur les jeux de l'Homme et de la Levure. Néanmoins, il apparaît dans certains cas que des couples de compartiments mènent à l'estimation de modèles suffisamment proches en variation totale pour toujours être regroupés en premier. C'est le cas du chloroplaste et de la mi-

tochondrie sur le jeu Swissprot, mais aussi du réticulum endoplasmique et du golgi sur le jeu de l'Homme.

	vraisemblance		Arbres			SVM
	ordre 0	ordre 1	ordre 0	ordre 1	proba o1	P.& K. (2003)
cytop	18 (27)	26 (40)	16 (44)	45 (51)	30 (40)	72.2
cytos	55 (5)	73 (1)	75 (10)	65 (40)	80 (50)	58.5
er	35 (12)	12 (13)	35 (18)	51 (63)	50 (60)	46.5
golgi	16 (2)	0 ( )	27 (7)	11 (41)	30 (50)	14.6
lyso	27 (6)	24 (14)	50 (21)	76 (39)	80 (40)	61.8
mito	13 (20)	20 (24)	41 (32)	44 (47)	50 (60)	57.4
nuc	26 (83)	23 (78)	64 (77)	76 (78)	80 (80)	89.6
pero	42 (5)	22 (20)	1 (6)	36 (43)	50 (50)	25.2
vacu	6 (1)	14 (14)	4 (9)	8 (57)	7 (30)	25.0
chlo	40 (26)	14 (48)	65 (31)	60 (50)	70 (60)	72.3
plas	78 (86)	52 (69)	86 (96)	95 (77)	100 (80)	92.2
extr	14 (54)	32 (62)	44 (64)	50 (71)	50 (70)	78.0
totale	35	30	54	66	68	78.2
moyenne	30	26	45	51	56	57.9

TAB. 7.8 – Résultats sur le jeu de Park et Kanehisa. Pour chaque compartiment et chaque méthode, la sensibilité (spécificité) est reportée. la colonne proba o1 donne les résultats obtenus pour un modèle d'ordre 1 en fixant un seuil sur la probabilité d'appartenance à chaque noeud (la couverture est de 87%).

Les tableaux 7.8, 7.9 et 7.10 présentent les résultats de classification obtenus respectivement sur les jeux de Park & Kanehisa, de l'Homme et enfin de la Levure. Les mesures de performances données ont été à chaque fois estimées par validation croisée 10 fois. Les classifications par arbre ont été réalisées à partir d'un arbre construit et possédant le même ordre de modèle. En raison de temps de calcul importants (on doit estimer autant de CMC que de noeuds et feuilles de l'arbre), les arbres ont été construits à partir du jeu de séquences complet.

Dans le cas de la levure, les résultats présentés sont comparés avec ceux obtenus par des SVM entraînés sur les fréquences des acides aminés et des codons. On a aussi rajouté sur une colonne les résultats publiés pour la version de PSORTII développée pour la Levure.

Examinons dans un premier temps la différence de performance attribuable à la méthode par arbre par rapport à une classification utilisant les vraisemblances. Pour tous les jeux de données, et tous les ordres de modèle, l'utilisation des arbres permet d'améliorer la sensibilité globale de 20 à 30 % (de 18% en ordre 1 sur la Levure à 36% en ordre sur le jeu Swissprot). Aussi, à l'exception de la Levure, la sensibilité

	vraisemblance		Arbres			Motifs
	order 0	order 1	order 0	order 1	proba o1	S.& al (2004b)
cytop	19 (23)	20 (23)	32 (41)	39 (38)	40 (40)	65 (68)
er	19 (29)	35 (24)	47 (31)	38 (38)	30 (50)	69 (83)
golgi	14 (10)	16 (10)	30 (9)	5 (10)	10 (30)	60 (74)
lyso	31 (8)	6 (4)	6 (11)	51 (23)	20 (60)	60 (71)
mito	43 (28)	41 (37)	49 (48)	62 (44)	80 (50)	67 (61)
nuc	28 (77)	38 (65)	61 (74)	72 (69)	80 (70)	93 (84)
pero	26 (2)	17 (16)	3 (3)	20 (100)	60 (100)	43 (50)
plas	19 (26)	34 (24)	9 (51)	36 (49)	40 (60)	89 (77)
extr	15 (38)	22 (33)	52 (49)	33 (63)	60 (60)	89 (87)
total	24	30	50	48	58	78
mean	23	25	38	40	47	70

TAB. 7.9 – Résultats sur le jeu de l'Homme. Mêmes notations que pour la table 7.8. la couverture pour la colonne proba o1 est de 78% (couverture de 74% dans [STH04])

moyenne augmente de 10 à 20%. Cette première constatation justifie donc dans le cadre d'une classification fondée sur une modélisation des séquences par CMC, la stratégie employée.

Sur le jeu Swissprot, les meilleurs résultats en sensibilité sont obtenus pour un arbre d'ordre 1 (66% de sensibilité globale). Ces résultats sont 10% en dessous de la sensibilité publiée dans [PK03]. Cependant, on peut remarquer que, malgré l'utilisation d'un modèle simple sur les séquences, la sensibilité pour les protéines de la membrane plasmique est de 95% (77 % de spécificité), et atteint 100% après seuillage sur les probabilités (80% de spécificité). Dans le cas des compartiments de faible effectifs, l'utilisation d'un modèle d'ordre 1 fait chuter la sensibilité mais augmente la spécificité. C'est par exemple le cas du golgi et du cytosquelette. Même si les performances sont améliorées pour certains compartiments, le seuillage sur les probabilités d'appartenance aux noeuds n'est pas particulièrement probant en terme de vraisemblance globale. On peut d'ailleurs remarquer que la sensibilité sur les protéines du cytoplasme chute de 15%. Ce compartiment abritant des catégories fonctionnelles très diverses, ceci pourrait être expliqué par un pouvoir descripteur insuffisant avec une seule CMC.

Les résultats sur le jeu de l'Homme, même s'il ne peuvent soutenir la comparaison avec les performances présentées à l'aide des méthodes par motifs, peuvent être considérés comme encourageants par leur réaction au seuillage sur les probabilités. Après seuillage, la sensibilité globale augmente en effet de 10% (7% pour la sensibilité moyenne). Remarquons qu'on obtient des sensibilités de 80% pour les protéines du noyau et de la mitochondrie, même si la spécificité reste faible dans le cas de la mitochondrie (50%).

Par contre, sur les protéines de la membrane plasmique, réputées faciles à clas-

	Vraisemblance		Arbres			SVM		PSORTII
	order 0	order 1	order 0	order 1	proba o1	AA	codons	N.&H. (1999)
cytop	50 (25)	26 (30)	42 (29)	51 (43)	30 (40)	46 (40)	57 (59)	67
cytos	26 (13)	25 (14)	47 (17)	20 (41)	30 (40)	23 (28)	14 (20)	1
er	51 (32)	42 (34)	54 (41)	33 (34)	40 (40)	43 (33)	41 (50)	35
golgi	24 (6)	11 (12)	7 (14)	24 (19)	20 (20)	3 (6)	9 (14)	0
mito	22 (54)	53 (43)	48 (55)	36 (67)	50 (60)	52 (57)	57 (59)	50
nuc	28 (64)	45 (71)	59 (69)	81 (67)	80 (70)	65 (62)	75 (63)	71
pero	8 (1)	12 (21)	0 (0)	16 (100)	60 (90)	12 (16)	4 (7)	35
vac	6 (7)	10 (10)	4 (18)	18 (21)	20 (40)	8 (9)	6 (8)	0
plas	13 (30)	44 (16)	29 (55)	27 (19)	30 (30)	33 (39)	31 (34)	63
total	29	40	48	54	61	50	55	57
mean	25	30	32	34	40	32	33	36

TAB. 7.10 – Résultats sur le jeu de la Levure. Même notations que pour la table 7.8. La couverture pour la colonne proba o1 est de 82%. les paramètres du SVM sont $c = 30000, g = 64$ pour les AA et $c = 128, g = 64$ pour les codons.

ser, en particulier avec des CMC, on n'obtient plus qu'une sensibilité de 36% à 40%. Ceci pourrait être expliqué par le fait que les protéines annotées comme membranaires dans Swissprot ne sont pas nécessairement des protéines de la membrane plasmique. A contrario, les protéines annotées comme appartenant par exemple au réticulum endoplasmique ou au golgi par Hera peuvent être des protéines membranaires.

Pour le jeu de la Levure, comparons dans un premier temps les résultats par arbre aux résultats obtenus à l'aide de SVM. Les performances de l'arbre d'ordre 1 sont ainsi légèrement meilleures que les résultats par SVM sur les fréquences des acides aminés (54% de sensibilité globale pour l'arbre d'ordre 1 et 50% pour les SVM). Celà est majoritairement dû à une meilleure classification sur les protéines du noyau (81% contre 65%) et du cytoplasme. Par contre, l'arbre discrimine moins bien les protéines du réticulum endoplasmique (33% contre 43%) et de la mitochondrie (36% contre 52%). En seuillant sur les probabilités d'appartenance à un compartiment, ces sensibilités deviennent comparables et la sensibilité globale est alors de 61% (couverture de 82%). Comme pour le jeu de séquences de l'Homme, la sensibilité sur la membrane plasmique est basse (30%).

La sensibilité globale de PSORTII est supérieure de 3% à la méthode par arbres. Remarquons cependant que PSORTII possède une sensibilité nulle sur les protéines du golgi et de la vacuole (1% sur le cytosquelette).

Ainsi, même si les performances des méthodes par arbres n'atteignent pas celles publiées, ce classificateur est au moins aussi précis qu'un SVM construit sur des fréquences d'acides aminés. Bien que les CMC servant à la classification utilisent un plus grand nombre de descripteurs, ce résultat est encourageant connaissant les faiblesses in-

hérantes aux méthodes fondées sur une minimisation du risque bayésien (section 3.2).

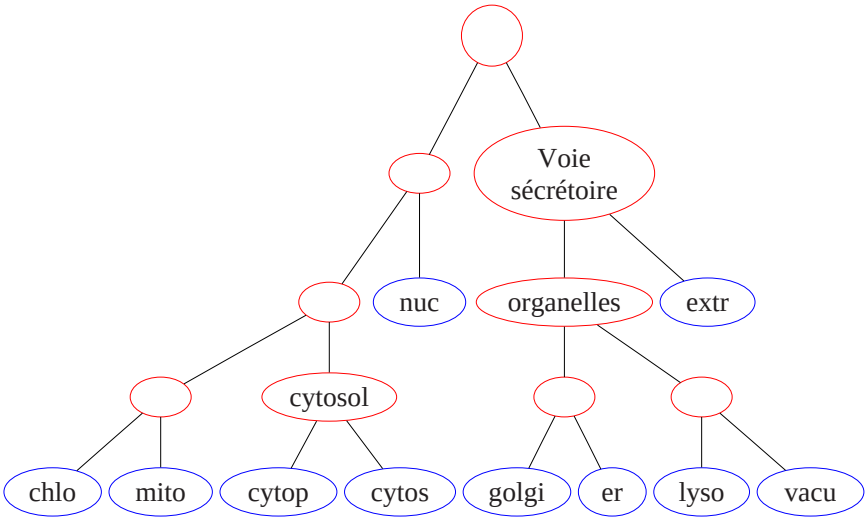


FIG. 7.9 – Arbre mimant le phénomène biologique d'adressage d'une cellule, et reprenant les grandes subdivisions de la méthode LOCtree. Les 6 compartiments considérés pour la classification sont donc le noyau, le cytosol, la mitochondrie, le chloroplaste, la matrice extracellulaire et les organelles de la voie sécrétoire. La membrane plasmique et le péroxisome ont été retirés lors de ces expériences.

Comme on a pu le constater, les performances de classification sont particulièrement mauvaises pour les compartiments de faible effectif. Nous nous sommes donc inspirés de la méthode LOCTree en construisant un arbre mimant le processus biologique d'adressage des protéines dans la cellule.

La figure 7.9 présente l'arbre créé pour le jeu de Park. Les arbres correspondants pour la Levure et l'Homme ont été construits sur la même base, en enlevant les protéines n'étant pas dans le jeu correspondant. On classe ainsi sur 6 compartiments pour le jeu de Park, et 5 pour l'Homme et 4 pour la Levure. A titre de comparaison avec les méthodes proposant la classification sur les mêmes compartiments, les protéines du péroxisome et de la membrane plasmique ont été retirées.

La table 7.11 compare les résultats obtenus sur nos trois jeux de données à ceux publiés dans [NR05] pour le jeu de test construit à partir de Swissprot version 41. Bien que la réduction des protéines homologues n'ait pas été menée avec autant de soin dans le cas des jeux de Park et de l'Homme, ces résultats fournissent une estimation moins biaisée pour ces classificateurs.

Les sensibilités globales obtenues sur les différents jeux sont d'un niveau comparable à celles reportées pour SubLoc, PSORTII et NNPSL. La meilleure sensibilité, obtenue avec le jeu de Park (67%) est néanmoins inférieure de 10% à celle reportée

						Arbres		
		LOCtree	SubLoc	PSORT II	NNPSL	Park	Homme	Levure
Voie Sécrétoire	sens.	87				81	78	60
	spec.	90				72	71	61
Organelles	sens.	51				46	66	60
	spec.	52				42	52	61
Extracell.	sens.	86	73	91	62	77	58	-
	spec.	93	53	32	63	67	70	-
Noyau	sens.	77	64	56	67	79	69	79
	spec.	85	71	75	59	78	75	74
Cytosol	sens.	82	43	47	42	49	46	23
	spec.	64	56	47	38	56	42	42
Mitochondrie	sens.	73	48	46	30	48	42	68
	spec.	78	59	59	67	49	54	55
Sensibilité globale		78	57	51	52	67	59	65

TAB. 7.11 – Comparatifs des résultats de classification pour l'arbre présenté sur la figure 7.9. Les résultats donnés sur les 4 premières colonnes reportent les performances sur le jeu de test construit à partir de Swissprot 41 de la table 5.3). La classification par arbres a été effectué avec des modèles d'ordre 1. Les performances sur les protéines du chloroplaste sont de 57% de sensibilité (59 % de spécificité) sur le jeu de Park (LOCTree : 77% et 63%).

pour LOCtree. A l'exception de PSORTII sur les protéines extracellulaires, les sensibilités/spécificités obtenues sur les protéines de chaque compartiments sont supérieures à celles de ces trois prédicteurs. Remarquons aussi une sensibilité de 66% pour le jeu de l'homme sur les organelles (51% pour LOCTree), mais au détriment des autres compartiments.

Ainsi si les résultats sont encourageants avec un nombre réduit de compartiments, ils nécessiteraient une vérification sur les jeux utilisés pour l'entraînement et le test du programme LOCtree.

Remarquons aussi que les auteurs de LOCtree reportent avoir augmenté leur sensibilité globale de 7% en intégrant des profils pour chaque protéine. L'intégration de ce type d'information à l'estimation des CMC, en remplaçant chaque protéine par son profil obtenu par alignement multiple, pourrait nous permettre d'améliorer nos résultats, en réduisant les problèmes posés par le surajustement.

Enfin, eu égard aux noyaux probabilistes présentés en section 3.3.4.2, on pourrait remplacer les étapes de classification par vraisemblance à chaque noeud par un SVM sur des noyaux de Fisher ou TOP. Toutefois, les premiers tests menés avec ces noyaux ne sont pas concluants.

Bibliographie

[CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM : a library for support vector machines*, 2001.

[DG00] A. Drawid and M. Gerstein. A bayesian system integrating expression data with sequence patterns for localizing proteins : comprehensive application for the yeast genome. *Journal Of Molecular Biology*, 301 :1059–1075, 2000.

[DJG02] A. Drawid, R Jansen, and M Gerstein. Genome-wide analysis relating expression level with protein subcellular localization. *Trends in Genetics*, 16(10) :426–430, 2002.

[ENBvH00] Olof Emanuelsson, Henrik Nielsen, Søren Brunak, and Gunnar von Heijne. Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. *J. Mol. Biol.*, 300 :1005–1016, 2000.

[HFG+03] Won-Ki Huh, James V. Falvo, Luke C. Gerke, Adam S. Carroll, Russell W. Howson, Jonathan S. Weissman, and Erin K. O'Shea. Global analysis of protein localization in budding yeast. *Nature*, 425 :686–691, October 2003. 10.1038/nature02026.

[HL04] Ying Huang and Yanda Li. Prediction of protein subcellular locations using fuzzy k-NN method. *Bioinformatics*, 20(1) :21–28, 2004.

[Ike85] T Ikemura. Codon usage and tRNA content in unicellular and multicellular organisms. *Mol Biol Evol*, 2(1) :13–34, 1985.

[NBM+02] Pierre Nicolas, Laurent Bize, Florence Muri, Mark Hoebeke, Francois Rodolphe, S. Dusko Ehrlich, Bernard Prum, and Philippe Bessieres. Mining Bacillus subtilis chromosome heterogeneities using hidden Markov models. *Nucl. Acids. Res.*, 30(6) :1418–1426, 2002.

[NEBvH97] Henrik Nielsen, Jacob Engelbrecht, Søren Brunak, and Gunnar von Heijne. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering*, 10 :1–6, 1997.

[NR05] Rajesh Nair and Burkhard Rost. Mimicking cellular sorting improves prediction of subcellular localization. *Journal of Molecular Biology*, 348 :85–100, April 2005.

[OS98] Matej Oresic and David Shalloway. Specific correlations between relative synonymous codon usage and protein secondary structure. *Journal of Molecular Biology*, 1998.

[PK03] Keun-Joon Park and Minoru Kanehisa. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19(13) :1656–1663, 2003.

[PL88] WR Pearson and DJ Lipman. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci.*, 85(8) :2444–2448, 1988.

[SL87] PM Sharp and WH Li. The codon Adaptation Index–a measure of directional synonymous codon usage bias, and its potential applications. *Nucl. Acids Res.*, 15(3) :1281–1295, 1987.

[SLHT04] M. Scott, G. Lu, M. Hallett, and D. Y. Thomas. The Hera database and its use in the characterization of endoplasmic reticulum proteins. *Bioinformatics*, 20(6) :937–944, 2004.

[STH04] Michelle S. Scott, David Y. Thomas, and Michael T. Hallett. Predicting Subcellular Localization via Protein Motif Co-Occurrence. *Genome Res.*, 14(10a) :1957–1966, 2004.

[Vap95] V. Vapnik. *The nature of Statistical Learning Theory*. Springer, New-York, 1995.

[Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley, New-York, 1998.

[Yua99] Z Yuan. Prediction of protein subcellular locations using markov chain models. *FEBS letter*, 451 :23–26, 1999.

Annexe A

Identification of programmed translational -1 frameshifting sites in the genome of *Saccharomyces cerevisiae*

Cette dernière partie présente une étude réalisée en parallèle du travail exposé précédemment, à l'initiative de Michaël Bekaert et Jean Pierre Rousset, du laboratoire de génétique moléculaire de la traduction et qui travaillent sur les phènomènes de recodage dans les génomes.

L'article qui suit présente un système d'identification de gènes dont l'expression est contrôlée par un décalage de phase de lecture en -1, c'est à dire où le ribosome, lors de la traduction, change de phase de lecture sur l'ARN messager en décalant celle ci d'une position vers l'arrière. Ce phénomène peut être attribué à des signaux stimulateurs sur l'ARNm distincts du site de décalage lui-même.

Jacks et al (1988) ont les premiers proposé un modèle de glissement en tandem pour expliquer le décalage de phase de lecture en -1 du virus du sarcome de Rous. Selon ce modèle, le déphasage se produit sur un heptamère glissant en phase 0 : `X-XXY-YYZ`[1]. Lorsque le ribosome arrive sur cette séquence, les ARN de transfert glissent simultanément d'un nucléotide en arrière.

La particularité de l'approche proposée à la suite par rapport à celles existantes sur le sujet est que la recherche se fait ici sans a priori sur le mécanisme impliqué, *i.e.* sans modélisation sur l'heptamère glissant. En outre, Michaël a ensuite testé expérimentalement tous les candidats identifiés. Ainsi, après avoir extrait du génome de la levure les structures compatibles avec un phénomène de frameshift, les candidats sont sélectionnés sur la base de deux critères indépendants :

[1] `X` correspond à n'importe quel nucléotide, `Y` est un nucléotide à appariement faible et `Z` dépend de l'espèce

– la présence de motifs protéiques qui pourraient être associés à un événement de décalage de phase.
– La vraisemblance de la séquence possédant un décalage de phase par rapport à un modèle de chaînes de Markov cachées estimé sur des séquences codantes.

Remarquons qu'à la différence de la problématique présentée dans la partie précédente, on procède ici à une classification non supervisée sur des séquences, en ne disposant pas de jeu de contrôle.

Ce système sélectionne 186 régions candidates. Michaël a ensuite vérifié l'expression des ARNm et évalué l'efficacité de déphasage *in vivo*. 11 régions, sur les 55 testées, présentent effectivement un décalage de phase de lecture au moins 50 fois au dessus du bruit de fond.

Une extension en cours de ce travail est d'adapter les modèles construit à la détection des sites potentiels pour d'autres types de décalage, tel que le décalage de phase de lecture en +1 et la translecture.

Michaël Bekaert, Hugues Richard, Bernard Prum & Jean-Pierre Rousset.

Abstract

Frameshifting is a recoding event that allows the expression of two polypeptides from the same mRNA molecule. Most recoding events described so far are used by viruses and transposons to express their replicase protein. The very few number of cellular proteins known to be expressed by a -1 ribosomal frameshifting have been identified by chance. The goal of the present work was to set up such a systematic strategy, based on complementary bioinformatics, molecular biology and functional approaches, without a priori knowledge of the mechanism involved. Two independent methods were devised. The first looks for genomic regions in which two ORFs, each carrying a protein pattern, are in a frameshifted arrangement. The second uses Hidden Markov Models and likelihood in a two step approach. When this strategy was applied to the *Saccharomyces cerevisiae* genome, 189 candidate regions were found, of which 58 were further functionally investigated. Twenty eight of them expressed a full length mRNA covering the two ORFs and 11 showed a -1 frameshift efficiency 50-fold higher than background, some of which correspond to genes with known functions. From others ascomycetes, 4 frameshifted ORFs are found fully conserved. Strikingly, most of the candidates do not display a classical viral-like frameshift signal and would have escaped a search based on current models of frameshifting. These results strongly suggest that -1 frameshifting might be more widely distributed than previously thought.

A.1 Introduction

Sequencing programs, along with various projects in the pharmaceutical, agricultural, aquacultural, and forestry industries, are creating an explosion of DNA sequence data. With this abundance of data, there is a growing need for more effective tools and methods to extract vital information from raw DNA sequences. Algorithms for identifying protein coding regions and predicting complete genes are of particular importance. Since the early 1990s, a number of computer programs for eukaryotic gene identification have been developed : GENMARK (Borodovsky and McIninch 1993), FGENEH (Solovyev and Salamov 1997 ; Solovyev et al. 1994), GeneParser (Snyder and Stormo 1995), GeneWise (Birney et al. 1996), GenScan (Burge and Karlin 1997), and Procruts (Gelfand et al. 1996 ; Mironov et al. 1998). Most of these programs make use of sophisticated pattern recognition techniques, such as linear discriminant analyses, neural networks, or Hidden Markov models to identify coding regions. Some programs also make use of database sequences alignment methods, such as BLAST (Altschul et al. 1990), to further improve their predictions. Generally, these algorithms classify out of frame ORFs as either a sequencing error or a pseudogene signature (Harrison et al. 2002). Up to now only a few algorithms assign a frameshift as a possible regulatory process. However, frameshifting together with readthrough of stop codons and ribosome hopping, is part of the reprogrammed genetic decoding (“recoding”) events that allow expression of several polypeptides from the same mRNA (Gesteland et al. 1992). Although most of the recoding events described so far have been found in small autonomous genetic elements (Baranov et al. 2002 ; Baranov et al. 2003 ; Bekaert and Rousset 2005), a few cellular genes are known to be expressed by this mode of control (Namy et al. 2004), most of them have been found by chance.

Twenty years ago, Jacks and Varmus described the first programmed -1 ribosomal frameshifting, event from which they established the canonical model of the eukaryotic -1 frameshifting site (Jacks et al. 1988 ; Jacks and Varmus 1985). Today, several tens of viruses and one mouse nuclear gene (Manktelow et al. 2005 ; Shigemoto et al. 2001) have been identified as bearing such a -1 frameshifting site. A typical eukaryotic site contains a slippery heptamer, where both A- and P-site tRNAs slip by one nucleotide upstream, followed by a stimulatory structure (stem loop, or pseudoknot) downstream (Brierley et al. 1989). The slippery heptamer is separated from the stimulatory structure by a short sequence, the so-called spacer. Based on this model, studies have been undertaken to identify frameshifting sites in the nuclear genome of the yeast *Saccharomyces cerevisiae* (Hammell et al. 1999 ; Liphardt 1999). However, none of these made it possible to identify with certainty authentic expressed genes controlled by -1 frameshifting. Two reasons might be proposed to explain this situation : first, the model might not be precise enough, leading to the identification of too many false positive candidates (Bekaert et al. 2003) ; conversely the model might be too rigid, failing to identify true positive candidates. This would be the case, for example, if -1 frameshift could be di-

rected by a more “degenerated” structure, or by mechanisms that rely on other types of signals.

Although most translational recoding events are found in viruses and transposons, a few cellular genes have been identified that use this mode of expression (Namy et al. 2004). These genes are involved in a variety of biological processes and are sometimes subject to a self-regulatory mechanism. Recoding is also widely distributed between organisms ; it is thus likely that numerous novel recoded cellular genes are still to be discovered. However, the prediction of recoding sites from genomic databases is currently a difficult task. Since most recoding events generate a premature in-frame stop codon, this is generally categorized as an error by computer programs, leading to improper gene annotation. Bioinformatics strategies have been developed to identify recoded genes, based on the knowledge of the recoding mechanism (model-based approach). In this case, genomic sequences are searched for regions exhibiting an already known recoding signal. Such analyses have allowed the identification of several candidate recoded genes (Baranov et al. 2002 ; Hammell et al. 1999 ; Namy et al. 2003). These approaches suffer major drawbacks : an imprecise model leading to a high number of false positive candidates and too rigid a model failing to identify truly positive candidates. For this reason, we and others have undertaken to develop bioinformatics approaches that do not depend on models of recoding sites and can be performed without a priori knowledge of the mechanism involved (Harrison et al. 2002 ; Sato et al. 2003). These approaches seek genomic configurations compatible with recoding, such as two ORFs overlapping or separated by a unique stop codon. The high number of candidates is then filtered by secondary constraints (length, presence of protein motifs, etc.). Several candidate recoded genes have already been identified in yeast (Harrison et al. 2002 ; Namy et al. 2003) and in Drosophila (Sato et al. 2003) in this way. However, except for one study (Namy et al. 2003), no biological validation has been performed to assess whether the candidate regions actually induce recoding *in vivo*.

The goal of the present work was to set up a comprehensive strategy, based on complementary bioinformatics and molecular approaches, and on functional *in vivo* analyses, to identify -1 ribosomal frameshifting sites in cellular genomes, without a priori knowledge of the mechanism involved. We devised two independent methods to look for frameshifting sites *in silico*. The first is based on the search for genomic regions in which two domains, each carrying a protein pattern, can be associated on a same polypeptide by a single -1 frameshifting event. The second is performed by a two step selection with Hidden Markov Models. The first step identifies potential candidates likely to possess a constrained coding region after their stop codon. The second step ranks the candidates by likelihood ratio, based on available biological knowledge. These two approaches do not rely on any model of the frameshifting site and thus are well adapted for *de novo* detection of frameshift events.

We validated these methods by analyzing the genome of *S. cerevisiae*. A total of 189 frameshifted candidate regions (fsORF) were found. We assessed the presence of a full-

length mRNA and quantified -1 frameshift efficiency for a subset of the highest ranked candidates. Among the 58 characterized regions, 28 were analyzed for their ability to induce -1 frameshifting *in vivo* ; 11 showed a frameshift efficiency 50-fold higher than the background. Several of these candidates correspond to genes with known functions, which will allow further analysis of the physiological role of the frameshifting event. Overall, these results strongly suggest that -1 frameshift might be a more widely used strategy of controlling gene expression than previously thought.

A.2 Results

A.2.1 General strategy

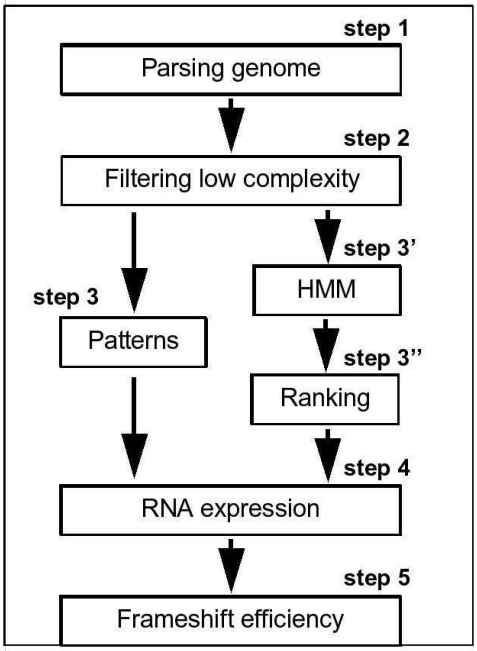


FIG. A.1 – Pipeline of frameshifting candidate identification strategy.

Figure A.1 shows the pipeline of our -1 identification strategy. We first download and parse the nucleic acid sequences, the intron/exon data and their position on chromosomes. We stock them in a local database for more reliability. Our system seeks genomic configurations compatible with a -1 ribosomal frameshifting event using the following criteria : two open reading frames, one in the 0 frame (ORF0), the other in the -1 frame (ORF-1) that overlap along an intermediate shared region (Step 1).

The second step was to filter undesirable low complexity sequences that may overload the next levels. The remaining sequences were classified according to whether the

0 and/or -1 frames are already annotated as an ORF, in order to perform the subsequent HMM step. We define four classes, "*left*" (ORF0 is annotated), "*right*" (ORF-1 is annotated), "*both*" (both ORFs are annotated) and "*none*" for all the others (Step 2). This classification is necessary, as the model with frameshift will be compared either to a coding one (if there is yet any annotation), or to a non coding model.

Two analyses were then carried out in parallel in Step 3 ; Regions that have protein patterns in both ORF0 and ORF-1 were retained. In parallel (Step 3' and Step 3"), Hidden Markov Models filtering and estimation was performed to predict coding regions that may continue in the -1 frame after the stop codon of ORF0. This was followed by a ranking step where we compared the likelihood ratio of each selected candidate structure on the two following assumptions : "the sequence possesses a frameshift", and, "the sequence does not possess any frameshift", taking into account the class of the candidate defined in Step 3.

We then tested the candidate regions for expression *in vivo*, by looking for the presence of a full-length polyadenylated mRNA, using oligo(dT) primed RT-PCR (Step 4). Finally, for the remaining candidates, -1 frameshifting efficiencies were determined *in vivo*, using a dual reporter system (Step 5).

A.2.2 Creating a dataset of potential -1 frameshift regions

The goal of this step was to identify structures exhibiting a genomic organization compatible with a translational -1 frameshift mode of expression. We chose to search first for overlapping ORFs. We fixed a length of at least 99 nucleotides for both ORF0 and ORF-1 areas, and at least 150 base pairs for the entire structure (figure A.2). Preliminary analysis (data not shown) had shown that decreasing this size by 2 fold (51 nucleotides) increased five times the numbers of retrieved structures. Thus, although biologically pertinent candidate might have been obtained with less stringent length constraints, this limit was chosen to keep the number of candidates compatible with the biological validation step. All searches were performed independently on four sets of data : the *S. cerevisiae* genome (12 Mbp), the genome of the yeast L-A virus (4,579 bp) known to bear an authentic -1 ribosomal frameshifting site, and artificial genomes that exhibit the same hexamer frequencies as the yeast and L-A genome respectively. The artificial genomes were generated using Markov chains (see Methods). The interest of these sequences was to generate negative controls to estimate, both quantitatively and qualitatively, the background or fortuitous candidates. All possible frameshifted structures were then automatically extracted. Among all potential -1 frameshifts, some are DNA microsatellites (Hamada et al. 1984) i.e. tandem repeats of the same triplet that are read as repetitions of two different amino acids, depending on the reading frame. Such sequences were excluded by using the mdust software, which removes low-complexity sequences. From this analysis 22,445 regions were found in the yeast genome, 24,248 in the artificial genome, 10 in the yeast L-A virus genome and 8 in the artificial L-A

genome.

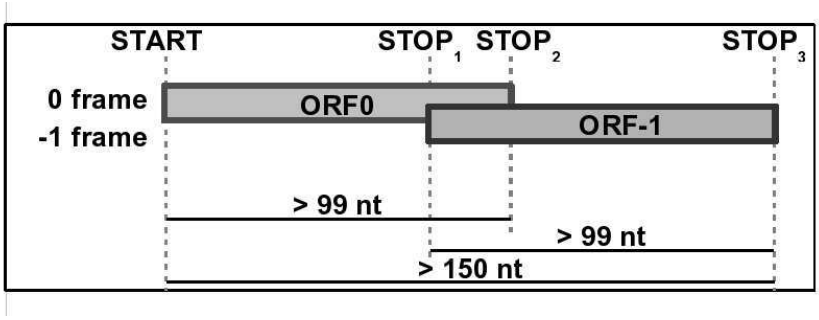


FIG. A.2 – Schematic representation of the genomic configurations compatible with -1 ribosomal frameshifting.

A.2.3 Assessing functional frameshifting by InterproScan

The all hit sequences were then subjected to protein motifs search. Each candidate sequence was kept only if it exhibited, in both frames, a pattern featured by the InterPro database and InterProScan. Since this step was the most time-consuming of the whole analysis, it was first performed on the smallest ORF. This database includes BlastProDom, FPrintScan, HMMPIR, HMMPfam, HMMSmart, HMMTigr, ProfileScan, ScanRegExp and SuperFamily. The default parameter settings were used for the search.

This approach was validated as far as only the actual frameshifting region was retrieved from the L-A virus genome. Moreover, 84 candidates were found in the yeast genome and only 11 in the yeast artificial genome. Among these 84 regions, three categories could be defined. In the first category, 69 exhibited domains that contain stretches of repeated amino acids in each of the two frames. These are not low-complexity sequences that were already discarded at Step 2 but correspond to an area with a high-density of an amino acid and not a linear repetition of the same amino acid. Noteworthy, no such candidates were found in the random genome. The second category is composed of regions in which the two ORFs bear similar protein patterns, or two distinct but functionally compatible motifs (e.g. a sugar transporter and a sugar binding site). We found 6 such regions in the yeast genome and none in the random genome. The third category includes 8 regions that bear functional regions in one ORF and amino acid repetitions in the other ORF. All the 11 candidate sequences from the random genome belong to this category.

A.2.4 Obtaining structure candidates by HMM

One of the more efficient methods to segment sequences in coding and non coding regions (allowing for different phases and genes on both strands) is the Hidden Markov Model (HMM). It was introduced by Rabiner for speech recognition (Rabiner 1989). This method is now commonly used in bioinformatics, from gene detection to prediction of protein domains (Burge and Karlin 1997 ; Nielsen et al. 1999 ; Sonnhammer et al. 1998).

For each step to be performed in a HMM framework, one has to completely specify a model, *i.e*. a probability law on the hidden states structure and a law for the emission of observed letters within each state. One has to note that the aim pursued here is not to *simply* to detect genes, but rather to select candidates for which the extension after the stop, in the -1 frame, is similar to coding regions. As far as we know, existing softwares designed for gene detection do not offer such flexibility : at present they are designed to detect non overlapping genes and are surely not able to detect a coding sequence with a frameshifting site. The beginning of such a gene may be missed if the length between the start codon and the frameshift is too short. Even when it is found, the program will probably decide on a false end, based on to the presence of a stop codon. In addition, the part after the frameshift will “never” be detected because of lack of start codon. Moreover, such software are established and tested on a sufficiently large set of labeled examples, which do not yet exist in the case of ribosomal frameshift. In the following paragraph, we detail the construction of the HMM and the strategy used for detection and ranking.

First, one needs to describe a model fitting with gene structure constraints. The simplest structure is summarized in figure A.3, and corresponds to the one used by common gene detectors (Burge and Karlin 1997). A gene begins with a start, continues by stretches of three bases corresponding to the codons and ends on a stop. Previous studies have demonstrated the distribution of codon - and thus amino acid - heterogeneities within genes (Nicolas et al. 2002). To take this type of heterogeneity into account, we allowed the model to alternate between up to three different laws for codons. All parameters of this model were first estimated on a similarity reduced set of 3,158 ORFs (see Methods for details).

Then, to adapt our model for the detection of frameshifted genes, we allowed coding regions to appear in the -1 frame after the stop. For this purpose, we inserted a transition from the state corresponding to the last base of the stop to the -1 coding frame of each coding type. We only kept those sequences for which the sum θ of the corresponding transition probabilities was higher than 0.95, which corresponds to the clearcut threshold shown in figure A.4.

As a positive control, we tested this step of our approach on the L-A virus. This virus is selected with a probability θ of 1.0 (this is only due to approximation errors), whereas the other candidates from the L-A virus, as the candidates from the yeast random

genome reach at most a probability of 0.5.

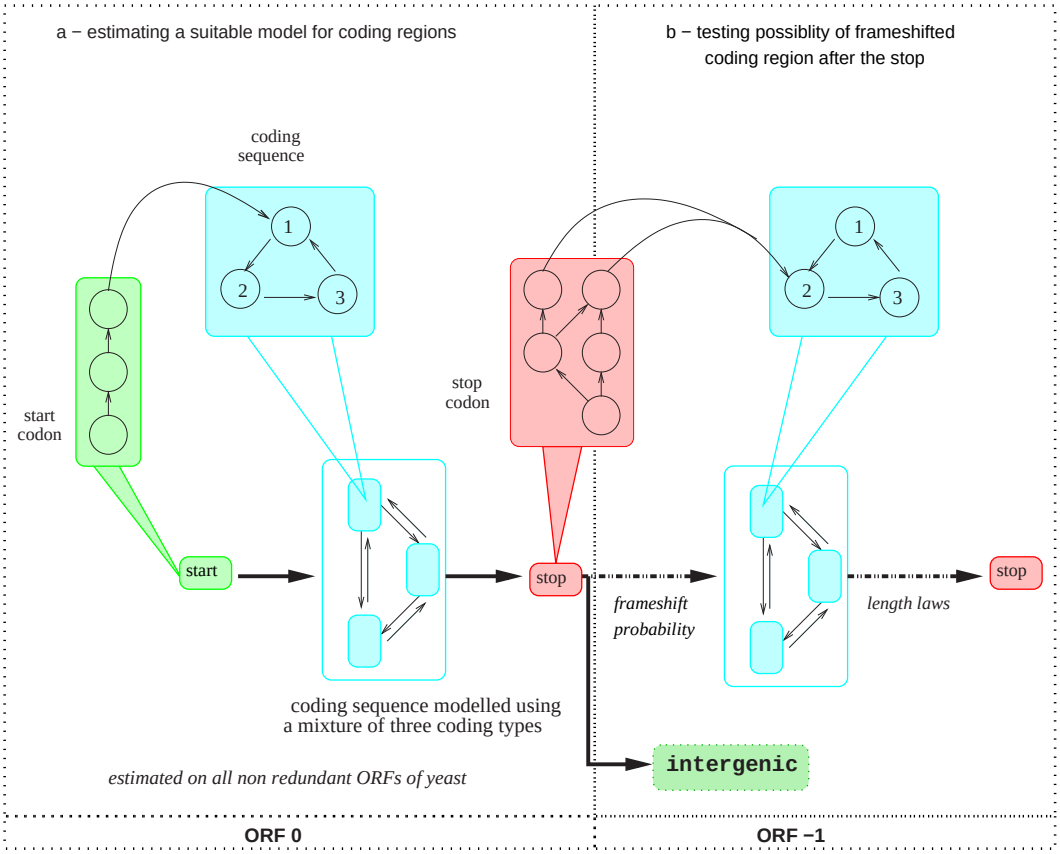


FIG. A.3 – Illustration of the HMM structures used for estimation and testing. (a) Estimating a suitable model for coding regions. The gene model estimated on all non-redundant ORFs of yeast is fitting. (b) Estimating additional parameters before filtering step, in order to test the possibility of frameshifted coding region after the first stop (represented by a dashed arrow).

Using this criterion, a final set of 110 candidates was retrieved. To incorporate for each selected candidate the known coding status of the two possible coding frames, we separately treated the sequences in the four classes defined above : *left*, *right*, *both* and *none*. In each class, we then ranked the sequences according to the likelihood ratio, which is a measure of the confidence we may assign to the claim “X contains a frameshift” in comparison with “X does not contain a frameshift” :

$$L_X = \frac{\mathbf{P}(X \mid \boldsymbol{\theta}_{\text{fs}}, S)}{\mathbf{P}(X \mid \boldsymbol{\theta}_{\text{nofs}}, S)}$$

Where $\boldsymbol{\theta}_{\text{fs}}$ and $\boldsymbol{\theta}_{\text{nofs}}$ stand respectively for the parameters of the model under the two following assumptions : “a frameshift exists” and, “no frameshift exists” conditionally

on the status of the ORF. More details about the models used conditionally on the subset can be found in the Methods section.

Candidates with their rank are summarized in Table A.1. From these scores, we selected 23 candidates to be tested (7 from *none* class, 7 from *both* class, 5 from *left* class and 4 from *right* class). Figure A.5 shows a representation of a "good" (fsORF 25) and a "bad" (fsORF 36) candidate.

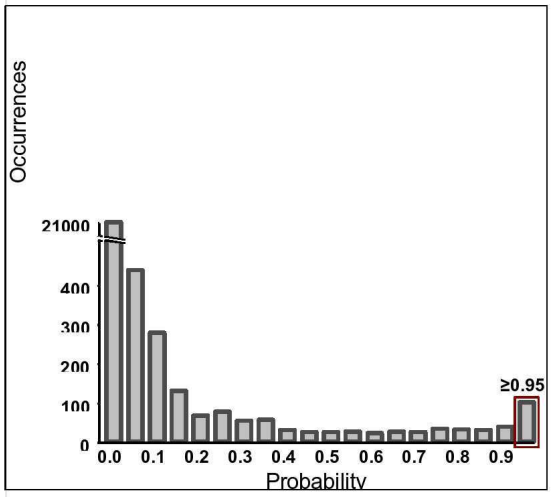


FIG. A.4 – Distribution of the probability of transition from the 0 to the -1 frame for the candidate regions compatible with a -1 frameshifting event. As evidenced in this distribution, a clear peak was observed at the 0.95 limit. This threshold value was thus chosen as a cutoff to choose the candidates to be ranked.

A.2.5 Common candidates

Finally, we crossed the results obtained using the protein motifs search and the HMM search. 5 common candidates were identified by comparing the 84 regions obtained in the first approach with the 110 regions obtained in the second approach. As the two methods are independent, these 5 common candidates together with 25 candidates from the protein motifs approach and the 18 best ranked candidates from the HMM approach were selected for further biological investigation. We also selected the 10 worse candidates to serve as a control of the relevance of the ranking procedure (table A.1).

A.2.6 Genomic sequence of the candidates

Since an authentic frameshifting is indistinguishable from a sequencing error, we first verified the sequence of the genomic region spanning the overlap between ORF0

Pattern Results

fsORF	Chr.	Location	gDNA[a]	mRNA[b]	cDNA[c]	FS	Class
1*[d]	I	192541-196178	+1 nt	-	-	-	both
2	**II**	**289386-290383**	**yes**	**yes**	**yes**	**6.0±1**	**left**
3[e]	II	454780-457622	yes	yes	yes	1.1±1	left
5	II	701799-700347	yes	no	-	-	left
6*	III	200170-197617	yes	yes	yes	0.1±0	both
10*	IV	167806-164992	yes	yes	yes	3.0±0	both
14[e]	IV	809035-808330	yes	yes	yes	1.8±0	none
15[e]	IV	890828-890321	yes	no	-	-	none
17*	V	298948-301706	yes	no	-	-	left
19c	**VI**	**15473-14309**	**yes**	**yes**	**yes**	**9.0±1**	**both**
20*	VII	1068995-1067213	yes	no	-	-	left
22*	VII	270340-267730	yes	yes	yes	0.5±0	both
23	VII	425616-425971	yes	yes	yes	n.a	none
24	**VII**	**677871-678301**	**yes**	**yes**	**yes**	**13.0±1**	**none**
32[d]	XI	172169-171299	yes	yes	yes	0.1±0	left
34*	XI	549085-551003	+1 nt	-	-	-	left
37*	XII	200413-200654	yes	no	-	-	none
38	XII	203255-204786	yes	yes	yes	n.a	both
40*	XII	857539-861524	yes	no	-	-	both
42	XIII	349605-348426	yes	yes	yes	n.a	left
43*c	**XIII**	**436627-438788**	**yes**	**yes**	**yes**	**5.0±1**	**both**
44*	**XIII**	**509318-507416**	**yes**	**yes**	**yes**	**5.0±1**	**left**
45	XIII	623212-622159	no	-	-	-	left
46	**XIII**	**650035-651026**	**yes**	**yes**	**yes**	**10.0±1**	**left**
48	XIV	40618-42065	yes	no	-	-	left
51	**XV**	**1026837-1028101**	**yes**	**yes**	**yes**	**7.0±1**	**left**
52	**XV**	**742910-744210**	**yes**	**yes**	**yes**	**5.0±1**	**left**
53	XV	758330-759354	yes	no	-	-	left
56	XVI	117365-117062	yes	yes	yes	0.1±0	none
57	XVI	138830-139449	yes	yes	yes	3.0±1	left

HMM results

fsORF	Chr.	Location	gDNA	mRNA	cDNA	FS	Class	Rank
19[e]	**VI**	**15473-14309**	**yes**	**yes**	**yes**	**9.0±1**	**both**	**1**
29*[d]	X	405173-406968	+1 nt	-	-	-	both	2
28*	X	219713-217406	yes	yes	yes	2.1±0	both	3
12*	**IV**	**384077-381986**	**yes**	**yes**	**yes**	**11.0±1**	**both**	**4**
43*[e]	**XIII**	**436627-438788**	**yes**	**yes**	**yes**	**5.0±1**	**both**	**5**
18*	VI	123462-129904	yes	no	-	-	both	6
41*	XIII	263477-266754	yes	yes	yes	n.a	both	7
33	**XI**	**374144-374853**	**yes**	**yes**	**yes**	**12.0±1**	**left**	**1**
25	VIII	262554-262197	yes	yes	yes	0.1±0	left	2
4[d]	II	554266-553504	+1 nt	-	-	-	left	3
36[d]	XI	639597-638535	+1 nt	-	-	-	left	4
3[e]	II	454780-457622	yes	yes	yes	1.1±1	left	5
11	IV	205690-205988	yes	yes	yes	0.1±0	none	1
50	XIV	537790-538010	yes	yes	yes	0.8±0	none	2
27	VIII	499891-499585	yes	no	-	-	none	3
47	XIV	394359-394026	yes	yes	yes	n.a	none	4
54	XV	782222-782003	yes	no	-	-	none	5
14[e]	IV	809035-808330	yes	yes	yes	1.8±0	none	6
15[e]	IV	890828-890321	yes	no	-	-	none	7
31	X	74021-74610	yes	intron	-	-	right	1
55	XV	80639-81189	yes	intron	-	-	right	2
35	**XI**	**611160-611899**	**yes**	**yes**	**yes**	**7.±1**	**right**	**3**
13	IV	630075-630598	yes	intron	-	-	right	4
7*[f]	III	220178-218372	yes	no	-	-	right	
8[f]	III	222829-223097	yes	yes	yes	0.1±0	none	
9[f]	III	91686-91455	yes	no	-	-	none	
16[f]	V	183582-183327	yes	no	-	-	none	
21[f]	VII	146543-146769	yes	no	-	-	none	
26[f]	VIII	35126-34916	yes	no	-	-	none	
30[f]	X	732756-732555	yes	no	-	-	none	
39[f]	XII	767116-766933	yes	no	-	-	none	
49[f]	XIV	429214-428983	yes	no	-	-	none	
58[f]	XVI	935319-935028	yes	no	-	-	none	

TAB. A.1 – For each tested candidate, we have reported its location, examined the genomic DNA sequence, tested for the presence of an mRNA, and in some cases, analyzed a cDNA sequence, and experimentally evaluated -1 frameshifting (FS). For Hidden Markov Models (HMM) candidates, the calculated rank is also reported for each class. Selected candidates are in bold and are described in Table A.2. * RT-PCR was carried with two sets of primers, [a] "yes" reports the presence of the expected gDNA sequence, "no" reports the lack of amplification of the corresponding genomic region, and "+1" reports the presence of an additional nucleotide, leading to an in-frame structure spanning both ORF0 and ORF-1. [b] "yes" or "no" states the presence or absence of an mRNA spanning the two ORFs, respectively ; "intron" reports the presence of a previously unidentified intron. [c] "yes" states the presence of the expected cDNA sequence. [d] Reannotated by the Saccharomyces Genome Database (SGD). [e] Candidates retrieved by both the HMM and protein pattern searches. [f] Control

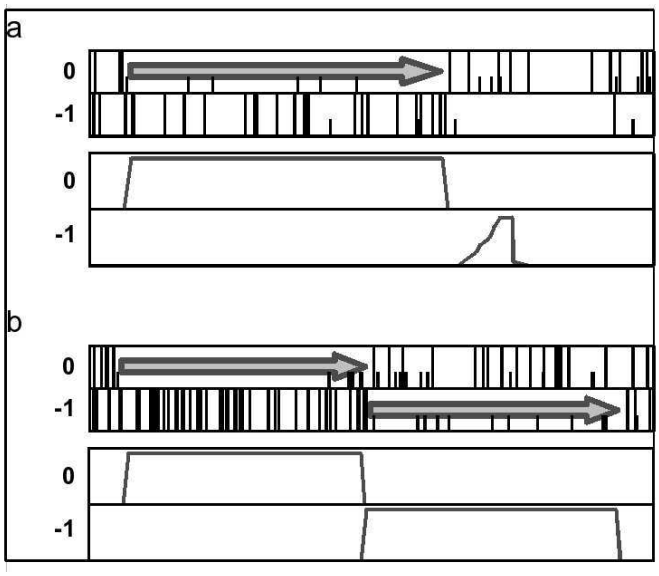


FIG. A.5 – A posteriori probabilities plot on the coding states on a "Bad" (fsORF 39) (a) and a "Good" (fsORF 28) (b) candidate from the *both* subset. On top, symbolical representation on the reading frames (plain-bars represent stop codon, half-bars initiation codon). On bottom, probability on coding on each frame. Arrows indicates coding frames.

and ORF-1. Among the 58 candidate sequences analyzed, 5 did not show the presence of the expected frameshift. Since the strain used here (FY1679-18B) is different from the strain that has been used for the *S. cerevisiae* sequencing project (S288C), either a sequencing/annotation error or a gene polymorphism could explain this discrepancy.

A.2.7 Expression of candidate sequences

The next step was to test whether the candidate sequences correspond to expressed ORFs. Since most of these regions were previously considered as intergenic region, they have not been included in systematic expression analyses. However, for those which are constituted of at least one previously annotated ORF (*right*, *left* and *both* classes), partial information was available and is indicated in Table A.1. However, even in the cases where the 2 ORFs were previously identified (*both* class), the presence of an mRNA corresponding to each ORF was tested independently. Thus, we checked whether an mRNA spanning the 2 ORFs is actually expressed. We examined the 53 remaining candidate sequences by RT-PCR, using first a reverse transcriptase step with an oligo(dT) primer that allows amplification of primarily polyadenylated mRNAs. The second PCR step was performed with an upper primer located 5' of the first ORF (0 phase) and a lower primer located near the stop codon in the second ORF (-1 phase) to ensure that a

full length message is actually present in the cell. For a few exceptionally long regions, a random primer was used in the reverse transcriptase step and two pairs of internal primers were used for the secondary PCR instead. No signal was observed in the absence of reverse transcriptase and a unique specific amplification was obtained for 31 candidate sequences (Figure A.2.7 and Table A.1). We retrieved 16 amplifications out of the 23 HMM candidates and 1 out of the 10 worse candidates from HMM controls ($\theta < 0.01$). The finding of much more putative frameshifting sites in the highly ranked candidates than in the lowest ranked candidates is a very strong argument in favor of their biological significance.

These results demonstrate that the same molecule of mRNA covers both ORFs and that these mRNAs are polyadenylated. The region of overlap of the cDNAs corresponding to all the bicistronic mRNAs was analyzed by gel electrophoresis and subsequently sequenced (data not shown). For 3 candidate regions, the presence of an unexpected intron was demonstrated (Table A.1). Close examination of the sequence revealed the regions harbor a degenerated intron boundary pattern. For the remaining candidates there was no evidence of length or sequence polymorphism, suggesting that no splicing or editing event had taken place.

fsORF	Level	Heptamer	Sage	Overlap	size (aa)	ORF0	ORF-1	Notes
2	6%±1	AAAAAAA	Low	34	332	SCO2		SCO2 (involved in stability of Cox1p and Cox2p)
12	11%±1	CCCAAAG	Low	64	698	YDL038C#	PRM7	PRM7 (pheromone-regulated membrane protein) •EC3.2.1.- : Glycosidases
19[e]	9%±1		-	145	389	AAD6	AAD16*	AAD6 (high similarity with the AAD of P. chrysosporium) •EC1.1.1.91 : Aryl-alcohol dehydrogenase (NADP+)
24	13%±1	UUUUUUU	-	88	143			Intergenic
33	12%±1		Medium	40	236	YKL033W-A#		-
35	7%±1		High	46	246		SRL3	SRL3 (Suppressor of Rad53 null Lethality)
43[e]	5%±1		-	43	720	YMR084W#	YMR085W#	putative glutamine–fructose-6-phosphate transaminase •EC2.6.1.16 : Glutamine–fructose-6-phosphate transaminase (isomerizing)
44	5%±1		Low	121	635	ADE17		ADE17 (AICAR transformylase/IMP cyclohydrolase) - Purine metabolism •EC2.1.2.3 : Phosphoribosylaminoimidazolecarboxamide formyltransferase •EC3.5.4.10 : IMP cyclohydrolase
46	10%±1		-	28	330	MRPL24		MRPL24 (Mitochondrial ribosomal protein)
51	7%±1		Low	49	421	RAD17		RAD17 (DNA damage checkpoint control protein)
52	5%±1		Low	199	433	STE4		STE4 (GTP-binding protein beta subunit of the pheromone pathway)

TAB. A.2 – fsORF with more than 5% of -1 frameshifting. (#) Hypothetical ORF

A.2.8 Quantification of -1 frameshift efficiency

It cannot be predicted whether ribosomes can actually shift from ORF0 to ORF-1 for 28 of these candidate expressed sequences, since none of them carries a canonical -1 frameshift signal, i.e. a heptamer followed by a secondary structure. To quantify -1 frameshift accurately, each fragment (about 50 nt either side of the overlapping areas) was amplified by PCR from genomic DNA of a wild-type yeast strain (FY1679-18B) and cloned into the pAC99 dual reporter vector (Namy et al. 2002). In this reporter system, each translating ribosome gives rise to β-galactosidase activity whereas only those

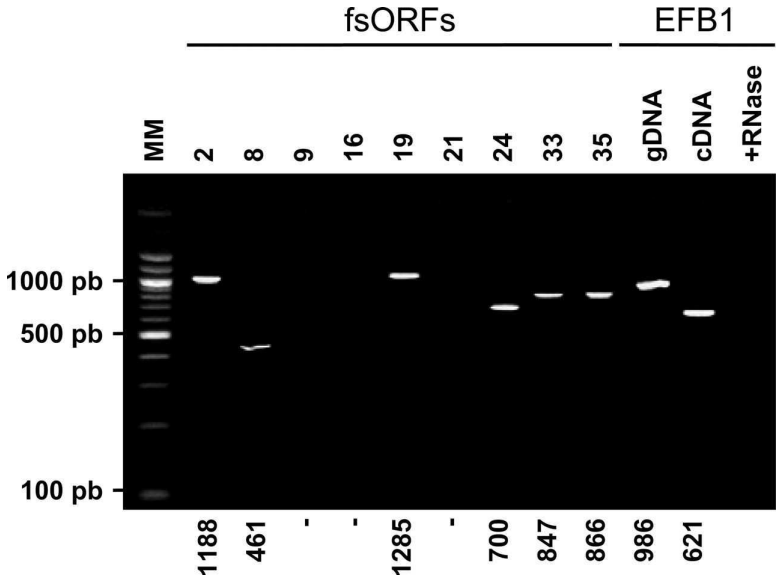


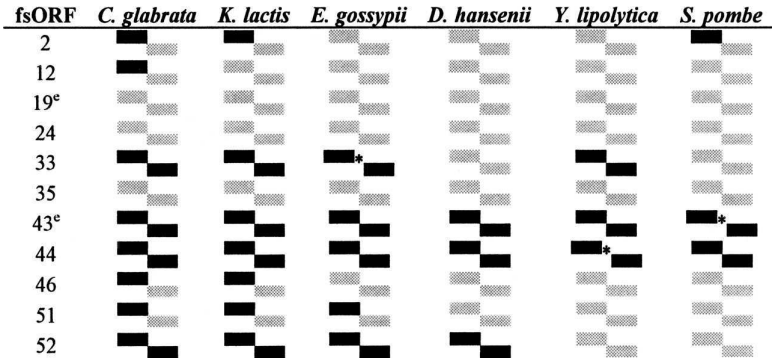
FIG. A.6 – RT-PCRs. Total RNA was extracted as described in Methods, and treated with DNAse I. RT-PCR was carried out in two steps. First, a reverse transcription was carried out using an oligo(dT) primer, allowing only reverse transcription of poly(A) mRNAs. Then a standard PCR was performed on the mRNA after the reverse transcription. The PCR products were visualized in a 1.5% agarose gel stained with ethidium bromide. A single amplification product was seen in all lanes, the expected size is indicated (in nucleotides) for each product at the bottom of the gel. The control sample was *EFB1* mRNA which includes an intron. Specific PCR on genomic DNA and cDNA exhibits two different products. Reverse transcription after RNase shows no DNA contamination during the process.

that frameshift into the overlapping region spanning ORF0 and ORF-1 would give rise to luciferase activity. Frameshifting efficiency is estimated by dividing the luciferase/β-galactosidase ratio obtained from the test construct by the corresponding ratio obtained from an in frame control construct (see Methods). Eleven fragments (Table A.2) displayed a -1 frameshift efficiency $\geqslant$50-fold over the background (0.1%).

A.2.9 Ascomycetes conservation

In order to seek if the frameshift organization of the eleven fragments directing frameshifting in vivo is preserved in other yeasts, we carried out alignments of the sequences against genomic sequences of ascomycetes. We found four structures where only ORF0 is conserved in several genomes (-∞ $\leqslant$ e-value $\leqslant$ 4.3x10-23), one where ORF0 is only present in *Candida glabrata* genome (fsORF 12, e-value = 5.1x10-13), two where no homolog could be found (Table A.3). Interestingly, 4 structures

(fsORF 33, 35, 44 and 52) are completely preserved (ORF0, ORF-1 and frameshifted organization). Surprisingly, fsORF 33 and 35 were reported to have a polymorphism (frameshift mutation) in S. cerevisiae and to present only one reading frame (Brachat et al. 2003 ; Cliften et al. 2003 ; Kellis et al. 2003). Very recently, these two ORFs have been re-annotated in frame. Although strain to strain polymorphism could account for this observation, our results showing that the frameshifted structure are conserved in other ascomycetes strongly suggest that the frameshift is biologically significant.



TAB. A.3 – Schematic profile of ORF0 and ORF-1 conservations in ascomycetes. The left bar indicats ORF0, the right bottom bar ORF-1. If the ORF is preserved, it is represented in black. [e]See legend for table A.1, *Frameshifting is not preserved (ORFs are separated).

A.3 Discussion

Here, we describe a comprehensive analysis of the *S. cerevisiae* genome which attempted to identify cellular recoding events occurring during translational -1 frameshifting. We developed a genomic approach, seeking genes with an extended coding potential, without prior constraint from existing ideas on the -1 frameshift mechanism.

In a first step, 22,445 genomic structures were extracted from the genome of *S. cerevisiae*. This value relies on two strong assumptions. First, we chose to collect only extensions of polypeptide but no premature ending, although biologically pertinent frameshifting events, such as in the *E. coli DnaX*, could lead to the synthesis of a shortened product (Tsuchihashi and Kornberg 1990). Second, we specified the minimal size of each ORF to 99 nucleotides (33 amino acids).

Our approach identified 189 candidates in the yeast genome. None of them had previously been found using a similar approach developed by Harrison and coworkers (Harrison et al. 2002). This study involved a pattern-based method, followed by a sequence comparison step against Genolevure, MIPS or SGD annotated ORFs. Neither

had any of our candidates have also been found by Hammell and coworkers, using a model-driven approach based on canonical frameshift signals.

Among the 189 candidate regions, 58 were analyzed further. Fifty of them showed the expected sequence of which 31 directed transcription of a mRNA spanning the two overlapping ORFs. These 28 regions were cloned in a dual reporter vector and 11 directed a -1 frameshifting efficiency 50-fold higher than background. To detect a possible mRNA editing mechanism, we sequenced the RT-PCR products for each of them. No RNA post-transcriptional modification was identified (Table A.2). Moreover, from the amplification of the mRNA using a poly(dT) primer in the reverse transcription step, we concluded that these mRNAs are polyadenylated and not rapidly degraded.

No candidate conformed to the canonical model of -1 frameshifting sites of Jacks and Varmus (Jacks et al. 1988). Three candidates exhibited a shifty heptamer in the appropriate frame but no detectable secondary structure (Table A.2). Others might correspond to a -1 frameshifting event carrying a more degenerate site or even correspond to a completely different mechanism but ending with an apparent -1 frameshift, such as ribosome hopping, +2 frameshifting or minority alternative splicing. In this latter case, the intron should be small since no differences in cDNA length were observed in the RT-PCR experiments. Some of these candidates might also turn out to be irrelevant with respect to frameshifting. In particular, some may correspond to pseudogenes or long 5' or 3' UTRs. Previous experiments have demonstrated that minimal frameshift signals from prokaryotic genomes can trigger ribosomes to shift to either the +1 or -1 frame *in vitro* (Gurvich et al. 2003). However, the same sequences in their genomic context failed to induce significant frameshifting, probably due to the sequence surrounding the frameshift site that may have evolved to suppress this phenomenon. Although this could apply for some candidates, we think this explanation is unlikely since our candidate sequences have been tested *in vivo* and with their surrounding sequence. Furthermore, during the last years, we have tested several dozens of constructs for basal frameshifting efficiency and found systematically a background value between 10^{-4} and 10^{-3}. A strong argument in favor of the biological significance of a subset of these putative frameshifted ORFs is their conservation in others ascomycetes. Among these four ORFs, the Rad17 gene seems particularly interesting since it plays a key role in monitoring the progress of DNA replication via its interaction with DNA polymerase ε (Post et al. 2003).

In conclusion, the combination of two simple approaches made it possible to identify several candidate genes potentially controlled by a -1 frameshift mechanism. This strategy is promising and could be straightforwardly extended to other organisms, eukaryotic as well as prokaryotic (Bertrand et al. 2002) and to other recoding events. Finally, we hope that the identification of new cellular recoded genes will also tell us whether they share similar properties or play common physiological roles in the cell.

A.4 Methods

Data sources

The system uses entire chromosome sequences from the GenBank/RefSeq database (Maglott et al. 2000) as inputs *S. cerevisiae* chromosomes NC_001133 to NC_001148 (downloaded on March 5, 2003) and *S. cerevisiae* virus L-A, NC_003754 (downloaded on December 25, 2003).

Random sequences

To define random background to be compared with real genome analyses, searches were performed independently on artificial genomes which exhibit the same hexamer frequencies as the *S. cerevisiae* genome or the L-A virus genome. We used the GenRGenS software v1.0 (Denise et al. 2003) for random generation of genomic sequences, using Markov chains of order 5.

Implementation

The main system is implemented in Perl, Bioperl 1.1 (Stajich et al. 2002) and PostgreSQL.

To detect protein signatures in the sequences, the motif database InterPro release 7.0 (Mulder et al. 2003) was used along with the software InterProScan version 3.1 (Zdobnov and Apweiler 2001).

In terms of family coverage, the protein signature databases are similar in size but differ in content. While all the methods share a common interest in protein sequence classification, some focus on divergent domains (e.g., Pfam), some focus on functional sites (e.g., PROSITE), and others focus on families, specializing in hierarchical definitions from superfamily down to subfamily levels in order to pin-point specific functions (e.g., PRINTS). TIGRFAMs focus on building HMMs for functionally equivalent proteins and PIR SuperFamilies, that produce HMMs over the full length of a protein and have protein length restrictions to gather family members. SUPERFAMILY is based on structure using the SCOP superfamilies as a basis for building HMMs. ProDom uses PSI-BLAST to find homologous domains that are clustered in the same ProDom entry. The clustered resources are derived automatically from the UniProt databases.

Low complexity filtering

The mdust algorithm (available from TIGR) was used to mask nucleic acid low-complexity regions, in particular from microsatellite areas, that enhance background noise and false positive.

Hidden Markov Models specification and estimation.

Each estimation and computation on Hidden Markov Models was done using the software SHOW (Nicolas et al. 2002). For the estimation of the coding parameters (defined as coding state ; figure A.3a), the ORF list of 5,861 sequences available on the SGD website was used. As *S. cerevisiae* is known to possess a large proportion of paralogous genes, we then wiped out proteins presenting more than 70% of full length similarity. All of these alignments were done using the FASTA program (Pearson 1990) using a BLOSUM62 matrix. Proteins were then clustered using a *p*-value threshold of 10^{-3}leading to a set of 3,526 sequences. The estimation of the intergenic state (composed of one state of order 2) was performed on the entire *S. cerevisiae* genome after masking of all of the annotated ORFs.

For the filter step (3'), the added links starting from the stop add 3 degrees of freedom to the model (the probabilities of shifting to the 3 possible coding states). In addition, 3 other parameters were added that correspond to the 3 coding states length laws from the STOP2 to the STOP3. We chose to estimate these 3 new length parameters only on the left, right and both subset. It was necessary to set up such a conservative fashion, since an important proportion of the 22,445 sequences considered could possibly influence the length estimation through an atypical composition in their intergenic regions. More precisely some intergenic regions appear to be better fitted by a mixture of two or three coding regions than by the intergenic law (figure A.3b). Probabilities of transition from the stop to the shifted coding regions were then deduced with a classical forward-backward algorithm on the 22,445 candidate structures to achieve step 3'.

For the ranking step, the likelihood of filtered sequences were calculated under the two assumptions : "the sequence contains a frameshift" and "the sequence contains no frameshift".

Whereas the first assumption corresponds to the same model for all of the candidates, different models were designed for each of the classes *left*, *right*, *none*, and *both* for the second assumption. These correspond to the following facts :

- none : "all the sequence is intergenic" ;
- left : "coding is followed by intergenic after STOP1" ;
- right : "coding ending on STOP3 is preceded by intergenic" ;
- both : "coding ends on STOP1, followed by intergenic and coding ending on STOP3".

The sequences were then ranked within each class on the log odd-ratio of the two concerned assumptions, rescaled by their length.

Ascomycetes comparison

FASTA (Altschul et al. 1990) was used for ascomycetes comparison. The FASTA search was executed (the e-value threshold was set to $1e^{-10}$) against the entire sequence

of the following genomes retrieved from GenBank/RefSeq database (Maglott et al. 2000) : *Candida glabrata* (NC_005967-68 & NC_006026-36), *Debaryomyces hansenii* (NC_006043-49), *Eremothecium gossypii* (NC_005782-88), *Kluyveromyces lactis* (NC_006037-42), *Schizosaccharomyces pombe* (NC_003421, NC_003423 & NC_003424) and *Yarrowia lipolytica* (NC_006067-72).

Yeast strains and media

The *S. cerevisiae* strain used for this work was FY1679-18B (Mat α *his3*-Δ200, *trp1*-Δ63, *ura3*-52, *leu2*-Δ1). The strain was grown in minimal medium (0.67% yeast nitrogen base, 2% glucose) supplemented with the appropriate amino acids to allow maintenance of the different plasmids under standard growth conditions. Yeast transformations were performed by the lithium acetate method (Ito et al. 1983).

Plasmids

The pAC99 reporter plasmid has been previously described (Namy et al. 2002). Constructs were obtained by inserting a PCR fragment containing the full overlapping region into the *MscI* cloning site, between the *lacZ* and *luc* genes in the plasmid pAC99. For -1 frameshift measurements, an in-frame control was used that allowed the production of 100% fusion protein (β-galactosidase-luciferase). The region including the inserted fragment was sequenced in the newly constructed plasmids. Each construct was then sequenced to check that no error occurred during PCR amplification.

Enzymatic activities and -1 frameshift efficiency

The yeast strains were transformed with the reporter plasmids using the lithium acetate method (Ito et al. 1983). In each case, at least five independent assays were performed in the same conditions. Cells were broken using acid-washed glass beads ; luciferase and β-galactosidase activities were assayed in the same crude extract, as previously described (Stahl et al. 1995). Efficiency of -1 frameshift is defined as the ratio of luciferase activity to β-galactosidase activity. To establish the relative activities of β-galactosidase and luciferase when expressed in equimolar amounts, the ratio of luciferase activity to β-galactosidase from an in-frame control plasmid was taken as a reference. Efficiency of -1 frameshift, expressed as percentage, was calculated by dividing the luciferase/β-galactosidase ratio obtained from each test construct by the same ratio obtained with the in-frame control construct.

Molecular biology procedures and RT-PCR

Each overlapping fragment corresponding to the candidate sequences was amplified from FY1679-18B genomic DNA by PCR, using *Pfu* polymerase (Promega), and

cloned into the pAC99 vector and checked by sequencing.

Total RNA was extracted from 5 ml of exponential yeast culture (Schmitt et al. 1990). Each RNA sample was subjected to digestion with 10 U of RNAse-free DNAse I (Boerhinger) at 37˚C for 1 h. DNAse I was inactivated by heating at 90˚C for 5 min, as recommended by the manufacturer. RNA was reverse-transcribed with oligo(dT) or random primer by Superscript II Kit (Invitrogen) for PCR amplification with Taq polymerase (Amersham) in a Primus thermocycler (MWG-Biotech). PCR fragments were visualized in a 1.5% agarose gel.

Acknowledgments

We are very grateful to Alain Denise, Jean-Paul Forest, Christine Froidevaux, Michel Termier and members of the GMT laboratory for stimulating discussions. We are especially grateful to Anne-Lise Haenni for critically reading the manuscript.

References

Altschul, S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. 1990. Basic local alignment search tool. *J Mol Biol* **215 :** 403-410.

Baranov, P.V., R.F. Gesteland, and J.F. Atkins. 2002. Recoding : translational bifurcations in gene expression. *Gene* **286 :** 187-201.

Baranov, P.V., R.F. Gesteland, and J.F. Atkins. 2002. Release factor 2 frameshifting sites in different bacteria. *EMBO Rep* **3 :** 373-377.

Baranov, P.V., O.L. Gurvich, A.W. Hammer, R.F. Gesteland, and J.F. Atkins. 2003. Recode 2003. *Nucleic Acids Res* **31 :** 87-89.

Bekaert, M., L. Bidou, A. Denise, G. Duchateau-Nguyen, J.P. Forest, C. Froidevaux, I. Hatin, J.P. Rousset, and M. Termier. 2003. Towards a computational model for -1 eukaryotic frameshifting sites. *Bioinformatics* **19 :** 327-335.

Bekaert, M. and J.P. Rousset. 2005. An extended signal involved in eukaryotic -1 frameshifting operates through modification of the E site tRNA. *Mol Cell* **17 :** 61-68.

Bertrand, C., M.F. Prere, R.F. Gesteland, J.F. Atkins, and O. Fayet. 2002. Influence of the stacking potential of the base 3’ of tandem shift codons on -1 ribosomal frameshifting used for gene expression. *Rna* **8 :** 16-28.

Birney, E., J.D. Thompson, and T.J. Gibson. 1996. PairWise and SearchWise : finding the optimal alignment in a simultaneous comparison of a protein profile against all DNA translation frames. *Nucleic Acids Res* **24 :** 2730-2739.

Borodovsky, M. and J. McIninch. 1993. Recognition of genes in DNA sequence with ambiguities. *Biosystems* **30 :** 161-171.

Brachat, S., F.S. Dietrich, S. Voegeli, Z. Zhang, L. Stuart, A. Lerch, K. Gates, T. Gaffney, and P. Philippsen. 2003. Reinvestigation of the Saccharomyces cerevisiae genome annotation by comparison to the genome of a related fungus : Ashbya gossypii. *Genome Biol* **4 :** R45.

Brierley, I., P. Digard, and S.C. Inglis. 1989. Characterization of an efficient coronavirus ribosomal frameshifting signal : requirement for an RNA pseudoknot. *Cell* **57 :** 537-547.
Burge, C. and S. Karlin. 1997. Prediction of complete gene structures in human genomic DNA. *J Mol Biol* **268 :** 78-94.
Cliften, P., P. Sudarsanam, A. Desikan, L. Fulton, B. Fulton, J. Majors, R. Waterston, B.A. Cohen, and M. Johnston. 2003. Finding functional features in Saccharomyces genomes by phylogenetic footprinting. *Science* **301 :** 71-76.
Denise, A., Y. Ponty, and M. Termier. 2003. Random generation of structured genomic sequences. In *Recomb'03*, Berlin.
Gelfand, M.S., A.A. Mironov, and P.A. Pevzner. 1996. Gene recognition via spliced sequence alignment. *Proc Natl Acad Sci U S A* **93 :** 9061-9066.
Gesteland, R.F., R.B. Weiss, and J.F. Atkins. 1992. Recoding : reprogrammed genetic decoding. *Science* **257 :** 1640-1641.
Gurvich, O.L., P.V. Baranov, J. Zhou, A.W. Hammer, R.F. Gesteland, and J.F. Atkins. 2003. Sequences that direct significant levels of frameshifting are frequent in coding regions of Escherichia coli. *Embo J* **22 :** 5941-5950.
Hamada, H., M.G. Petrino, T. Kakunaga, M. Seidman, and B.D. Stollar. 1984. Characterization of genomic poly(dT-dG).poly(dC-dA) sequences : structure, organization, and conformation. *Mol Cell Biol* **4 :** 2610-2621.
Hammell, A.B., R.C. Taylor, S.W. Peltz, and J.D. Dinman. 1999. Identification of putative programmed -1 ribosomal frameshift signals in large DNA databases. *Genome Res* **9 :** 417-427.
Harrison, P., A. Kumar, N. Lan, N. Echols, M. Snyder, and M. Gerstein. 2002. A small reservoir of disabled ORFs in the yeast genome and its implications for the dynamics of proteome evolution. *J Mol Biol* **316 :** 409-419.
Ito, H., Y. Fukuda, K. Murata, and A. Kimura. 1983. Transformation of intact yeast cells treated with alkali cations. *J Bacteriol* **153 :** 163-168.
Jacks, T., H.D. Madhani, F.R. Masiarz, and H.E. Varmus. 1988. Signals for ribosomal frameshifting in the Rous sarcoma virus gag-pol region. *Cell* **55 :** 447-458.
Jacks, T. and H.E. Varmus. 1985. Expression of the Rous sarcoma virus pol gene by ribosomal frameshifting. *Science* **230 :** 1237-1242.
Kellis, M., N. Patterson, M. Endrizzi, B. Birren, and E.S. Lander. 2003. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature* **423 :** 241-254.
Liphardt, J. 1999. The mechanism of -1 ribosomal frameshifting : experimental and theoretical analysis. Churchill College, Cambridge.
Maglott, D.R., K.S. Katz, H. Sicotte, and K.D. Pruitt. 2000. NCBI's LocusLink and RefSeq. *Nucleic Acids Res* **28 :** 126-128.
Manktelow, E., K. Shigemoto, and I. Brierley. 2005. Characterization of the frameshift signal of Edr, a mammalian example of programmed -1 ribosomal frameshifting. *Nu-*

cleic Acids Res **33 :** 1553-1563.
Mironov, A.A., M.A. Roytberg, P.A. Pevzner, and M.S. Gelfand. 1998. Performance-guarantee gene predictions via spliced alignment. *Genomics* **51 :** 332-339.
Mulder, N.J., R. Apweiler, T.K. Attwood, A. Bairoch, D. Barrell, A. Bateman, D. Binns, M. Biswas, P. Bradley, P. Bork, P. Bucher, R.R. Copley, E. Courcelle, U. Das, R. Durbin, L. Falquet, W. Fleischmann, S. Griffiths-Jones, D. Haft, N. Harte, N. Hulo, D. Kahn, A. Kanapin, M. Krestyaninova, R. Lopez, I. Letunic, D. Lonsdale, V. Silventoinen, S.E. Orchard, M. Pagni, D. Peyruc, C.P. Ponting, J.D. Selengut, F. Servant, C.J. Sigrist, R. Vaughan, and E.M. Zdobnov. 2003. The InterPro Database, 2003 brings increased coverage and new features. *Nucleic Acids Res* **31 :** 315-318.
Namy, O., G. Duchateau-Nguyen, I. Hatin, S. Hermann-Le Denmat, M. Termier, and J.P. Rousset. 2003. Identification of stop codon readthrough genes in Saccharomyces cerevisiae. *Nucleic Acids Res* **31 :** 2289-2296.
Namy, O., I. Hatin, G. Stahl, H. Liu, S. Barnay, L. Bidou, and J.P. Rousset. 2002. Gene overexpression as a tool for identifying new trans-acting factors involved in translation termination in Saccharomyces cerevisiae. *Genetics* **161 :** 585-594.
Namy, O., J.P. Rousset, S. Napthine, and I. Brierley. 2004. Reprogrammed genetic decoding in cellular gene expression. *Mol Cell* **13 :** 157-168.
Nicolas, P., L. Bize, F. Muri, M. Hoebeke, F. Rodolphe, S.D. Ehrlich, B. Prum, and P. Bessieres. 2002. Mining Bacillus subtilis chromosome heterogeneities using hidden Markov models. *Nucleic Acids Res* **30 :** 1418-1426.
Nielsen, H., S. Brunak, and G. von Heijne. 1999. Machine learning approaches for the prediction of signal peptides and other protein sorting signals. *Protein Eng* **12 :** 3-9.
Pearson, W.R. 1990. Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol* **183 :** 63-98.
Pearson, W.R. and D.J. Lipman. 1988. Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A* **85 :** 2444-2448.
Post, S.M., A.E. Tomkinson, and E.Y. Lee. 2003. The human checkpoint Rad protein Rad17 is chromatin-associated throughout the cell cycle, localizes to DNA replication sites, and interacts with DNA polymerase epsilon. *Nucleic Acids Res* **31 :** 5568-5575.
Rabiner, L.R. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. IEEE* **77 :** 257-285.
Sato, M., H. Umeki, R. Saito, A. Kanai, and M. Tomita. 2003. Computational analysis of stop codon readthrough in D.melanogaster. *Bioinformatics* **19 :** 1371-1380.
Schmitt, M.E., T.A. Brown, and B.L. Trumpower. 1990. A rapid and simple method for preparation of RNA from Saccharomyces cerevisiae. *Nucleic Acids Res* **18 :** 3091-3092.
Shigemoto, K., J. Brennan, E. Walls, C.J. Watson, D. Stott, P.W. Rigby, and A.D. Reith. 2001. Identification and characterisation of a developmentally regulated mammalian gene that utilises -1 programmed ribosomal frameshifting. *Nucleic Acids Res* **29 :** 4079-4088.
Snyder, E.E. and G.D. Stormo. 1995. Identification of protein coding regions in geno-

mic DNA. *J Mol Biol* **248 :** 1-18.
Solovyev, V. and A. Salamov. 1997. The Gene-Finder computer tools for analysis of human and model organisms genome sequences. *Proc Int Conf Intell Syst Mol Biol* **5 :** 294-302.
Solovyev, V.V., A.A. Salamov, and C.B. Lawrence. 1994. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Res* **22 :** 5156-5163.
Sonnhammer, E.L., G. von Heijne, and A. Krogh. 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. *Proc Int Conf Intell Syst Mol Biol* **6 :** 175-182.
Stahl, G., L. Bidou, J.P. Rousset, and M. Cassan. 1995. Versatile vectors to study recoding : conservation of rules between yeast and mammalian cells. *Nucleic Acids Res* **23 :** 1557-1560.
Stajich, J.E., D. Block, K. Boulez, S.E. Brenner, S.A. Chervitz, C. Dagdigian, G. Fuellen, J.G.R. Gilbert, I. Korf, H. Lapp, H. Lehvaslaiho, C. Matsalla, C.J. Mungall, B.I. Osborne, M.R. Pocock, P. Schattner, M. Senger, L.D. Stein, E. Stupka, M.D. Wilkinson, and E. Birney. 2002. The Bioperl Toolkit : Perl Modules for the Life Sciences. *Genome Res.* **12 :** 1611-1618.
Tsuchihashi, Z. and A. Kornberg. 1990. Translational frameshifting generates the gamma subunit of DNA polymerase III holoenzyme. *Proc Natl Acad Sci U S A* **87 :** 2516-2520.
Zdobnov, E.M. and R. Apweiler. 2001. InterProScan–an integration platform for the signature-recognition methods in InterPro. *Bioinformatics* **17 :** 847-848.

Web site references

`http://bioperl.org/` ; The Bioperl Project
`http://cbi.labri.fr/Genolevures/` ; Genolevure
`http://www.lri.fr/~denise/GenRGenS/` ; GenRGenS home page
`http://www.ebi.ac.uk/interpro/` ; InterPro database
`http://mips.gsf.de/genre/proj/yeast/` ; Munish information center for protein sequences (MIPS)
`http://www.ncbi.nlm.nih.gov/` ; National Center for Biotechnology Information (NCBI)
`http://www.yeastgenome.org/` ; Saccharomyces Genome Database (SGD)
`http://www-mig.jouy.inra.fr/ssb/SHOW/` ; Structured HOmogeneities Watcher (SHOW)

Annexe B

Articles, Posters et Communications

Articles

[1] Bekaert, M., Richard, H., Prum, B. and Rousset J.P. : Identification of programmed translationnal -1 frameshifting sites in the genome of Saccharomyces cerevisiae, *Genome Research*, vol. 15, 2005, p. 1411-1420.
[2] Miele, V., Bourguignon, P.Y., Robelin, D., Nuel, G., Richard, H. : seq++, analysing biological sequences with a range of Markov-related models (2004), *Bioinformatics*, Advance Acces.
[3] Richard, H. et Nuel, G. SPA : Simple web-tool to assess statistical significance of DNA patterns, *Nucl. Acids. Res.*, vol. 31, n. 13, 2003, p.3679-3681.
[4] Robelin, D., Richard, H. et Prum, B. SIC : A tool to detect short inverted segments in a biological sequence, *Nucl. Acids. Res.*, vol. 31, n. 13, 2003, p.3669-71.
[5] Robin, S., Daudin J.-J.,Richard, H., Sagot, M.-F. and Schbath, S. Occurrence probability of structured motifs in random sequences, *J. Comp. Biol.*, vol. 9, p. 761-773.

Posters et Communications

[6] Richard, H., Mucchielli M., Prum B., Képès F. Hidden Markov Models Hierarchical Classification for ab-initio prediction of Protein Subcellular Localization. ISMB'05, Detroit, june 2005, PLoS CB poster.
[7] Richard, H., Mucchielli M., Prum B., Képès F. Discrimination of the subcellular locations of the yeast proteins by their biological sequence. JOBIM 2004 (Journées Ouvertes Biologie Informatique Mathématique), Montréal, juin 2004, présentation courte.
[8] Richard, H., Mucchielli M., Prum B., Képès F. Discrimination of the subcellular locations of the yeast proteins by their biological sequence. ECCB'03 (European Conference on Computational BIology), Paris, september 2003, poster PSA_17.

Bibliographie

[AB72] R Ash and R Bishop. Monopoly as a markov process. *Mathematics Magazine*, 45 :26–29, January 1972.

[AOR98] M.A. Andrade, S.I. O'Donoghue, and B. Rost. Adaptation of protein surfaces to subcellular location. *Journal of Molecular Biology*, 276(517-525), 1998.

[Azz96] A. Azzalini. *Statistical Inference Based on the likelihood*, volume 68 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, 1996.

[Bil61] P. Billingsley. *Statistical Inference for Markov Processes*. Statistical research monographs. University of Chicago Press, 1961.

[BP66] L. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37 :1554–1563, 1966.

[BS00] J.F. Bonnans and A. Shapiro. Perturbation analysis of optimisation problems, 2000.

[BT82] G. Blom and D. Thorburn. How many random digits are required until given sequences are obtained ? *J. Appl. Prob.*, 19 :518–531, 1982.

[BTM+02] Hideo Bannai, Yoshinori Tamada, Osamu Maruyama, Kenta Nakai, and Satoru Miyano. Extensive feature detection of N-terminal protein sorting signals. *Bioinformatics*, 18(2) :298–305, 2002.

[CAPPE97] J. Cedano, P. Aloy, J.A. Perez-Pons, and Querol E. Relation between amino acid composition and cellular location of proteins. *Journal Of Molecular Biology*, 266(3) :594–600, 1997.

[CC02] Kuo-Chen Chou and Yu-Dong Cai. Using Functional Domain Composition and Support Vector Machines for Pr ediction of Protein Subcellular Location. *J. Biol. Chem.*, 277(48) :45765–45769, 2002.

[CC04] Yu-Dong Cai and Kuo-Chen Chou. Predicting subcellular localization of proteins in a hybridization space. *Bioinformatics*, 20(7) :1151–1156, 2004.

[Cho01] Kuo-Chen Chou. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins : Structure, Function, and Genetics*, 43 :246–255, 2001.

[CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM : a library for support vector machines*, 2001.

[CNR00] M. Cokol, R. Nair, and B. Rost. Finding nuclear localization signals. *EMBO Reports*, 1(15) :411–415, 2000.

[Cow91] R Cowan. Expected frequency of dna patterns using whittle's formula. *Journal of Applied Probability*, 28 :886–892, 1991.

[CSTEK02] Nello Cristianini, John Shawe-Taylor, André Elisseeff, and J. Kandola. *Advances in Neural Information Processing Systems 14*, chapter On Kernel-Target Alignment, pages 367–373. MIT Press, 2002.

[CVBM02] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1) :131–159, 2002.

[DBNvHB04] Jannick Dyrlov Bendtsen, Henrik Nielsen, Gunnar von Heijne, and Soren Brunak. Improved prediction of signal peptides : Signalp 3.0. *Journal of Molecular Biology*, 340 :783–795, July 2004.

[DG00] A. Drawid and M. Gerstein. A bayesian system integrating expression data with sequence patterns for localizing proteins : comprehensive application for the yeast genome. *Journal Of Molecular Biology*, 301 :1059–1075, 2000.

[DJG02] A. Drawid, R Jansen, and M Gerstein. Genome-wide analysis relating expression level with protein subcellular localization. *Trends in Genetics*, 16(10) :426–430, 2002.

[DSA03] K. Duan, Keerthi S.S, and Poo A.N. Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51 :41–59, 2003.

[EEvHC03] Olof Emanuelsson, Arne Elofsson, Gunnar von Heijne, and Susana Cristobal. In silico prediction of the peroxisomal proteome in fungi, plants and animals. *Journal of Molecular Biology*, 330 :443–456, July 2003.

[ENBvH00] Olof Emanuelsson, Henrik Nielsen, Søren Brunak, and Gunnar von Heijne. Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. *J. Mol. Biol.*, 300 :1005–1016, 2000.

[Fel68] W. Feller. *Introduction to Probability Theory*, volume I. Wiley, third edition, 1968.

[Fre71] D Freedman. *Markov chains*. Holden-Day series in probability and statistics. Holden Day, 1971.

[Gus97] Dan Gusfield. *Algorithms on Strings, Trees and Sequences : Computer Science and Computational Biology.* Cambridge University Press, 1997.

[HFG+03] Won-Ki Huh, James V. Falvo, Luke C. Gerke, Adam S. Carroll, Russell W. Howson, Jonathan S. Weissman, and Erin K. O'Shea. Global analysis of protein localization in budding yeast. *Nature*, 425 :686–691, October 2003. 10.1038/nature02026.

[HL04] Ying Huang and Yanda Li. Prediction of protein subcellular locations using fuzzy k-NN method. *Bioinformatics*, 20(1) :21–28, 2004.

[HN96] P Horton and K Nakai. A probabilistic classification system for predicting the cellular localization sites of proteins. *Proc Int Conf Intell Syst Mol Biol.*, 4 :109–115, 1996.

[HS01] Sujun Hua and Zhirong Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8) :721–728, 2001.

[IIS+04] Tadashi Imanishi, Takeshi Itoh, Yutaka Suzuki, Claire O'Donovan, Satoshi Fukuchi, Kanako O. Koyanagi, Roberto A. Barrero, Takuro Tamura, Yumi Yamaguchi-Kabata, Motohiko Tanino, Kei Yura, Satoru Miyazaki, Kazuho Ikeo, Keiichi Homma, Arek Kasprzyk, Tetsuo Nishikawa, Mika Hirakawa, Jean Thierry-Mieg, Danielle Thierry-Mieg, Jennifer Ashurst, Libin Jia, Mitsuteru Nakao, Michael A. Thomas, Nicola Mulder, Youla Karavidopoulou, Lihua Jin, Sangsoo Kim, Tomohiro Yasuda, Boris Lenhard, Eric Eveno, Yoshiyuki Suzuki, Chisato Yamasaki, Jun-ichi Takeda, Craig Gough, Phillip Hilton, Yasuyuki Fujii, Hiroaki Sakai, Susumu Tanaka, Clara Amid, Matthew Bellgard, Maria de Fatima Bonaldo, Hidemasa Bono, Susan K. Bromberg, Anthony J. Brookes, Elspeth Bruford, Piero Carninci, Claude Chelala, Christine Couillault, Sandro J. de Souza, Marie-Anne Debily, Marie-Dominique Devignes, Inna Dubchak, Toshinori Endo, Anne Estreicher, Eduardo Eyras, Kaoru Fukami-Kobayashi, Gopal R. Gopinath, Esther Graudens, Yoonsoo Hahn, Michael Han, Ze-Guang Han, Kousuke Hanada, Hideki Hanaoka, Erimi Harada, Katsuyuki Hashimoto, Ursula Hinz, Momoki Hirai, Teruyoshi Hishiki, Ian Hopkinson, Sandrine Imbeaud, Hidetoshi Inoko, Alexander Kanapin, Yayoi Kaneko, Takeya Kasukawa, Janet Kelso, Paul Kersey, Reiko Kikuno, Kouichi Kimura, Bernhard Korn, Vladimir Kuryshev, Izabela Makalowska, Takashi Makino, Shuhei Mano, Regine Mariage-Samson, Jun

Mashima, Hideo Matsuda, Hans-Werner Mewes, Shinsei Minoshima, Keiichi Nagai, Hideki Nagasaki, Naoki Nagata, Rajni Nigam, Osamu Ogasawara, Osamu Ohara, Masafumi Ohtsubo, Norihiro Okada, Toshihisa Okido, Satoshi Oota, Motonori Ota, Toshio Ota, Tetsuji Otsuki, Dominique Piatier-Tonneau, Annemarie Poustka, Shuang-Xi Ren, Naruya Saitou, Katsunaga Sakai, Shigetaka Sakamoto, Ryuichi Sakate, Ingo Schupp, Florence Servant, Stephen Sherry, Rie Shiba, Nobuyoshi Shimizu, Mary Shimoyama, Andrew J. Simpson, Bento Soares, Charles Steward, Makiko Suwa, Mami Suzuki, Aiko Takahashi, Gen Tamiya, Hiroshi Tanaka, Todd Taylor, Joseph D. Terwilliger, Per Unneberg, Vamsi Veeramachaneni, Shinya Watanabe, Laurens Wilming, Norikazu Yasuda, Hyang-Sook Yoo, Marvin Stodolsky, Wojciech Makalowski, Mitiko Go, Kenta Nakai, Toshihisa Takagi, Minoru Kanehisa, Yoshiyuki Sakaki, John Quackenbush, Yasushi Okazaki, Yoshihide Hayashizaki, Winston Hide, Ranajit Chakraborty, Ken Nishikawa, Hideaki Sugawara, Yoshio Tateno, Zhu Chen, Michio Oishi, Peter Tonellato, Rolf Apweiler, Kousaku Okubo, Lukas Wagner, Stefan Wiemann, Robert L. Strausberg, Takao Isogai, Charles Auffray, Nobuo Nomura, Takashi Gojobori, and Sumio Sugano. Integrative annotation of 21,037 human genes validated by full-length cdna clones. *PLoS Biology*, 2 :e162 EP –, June 2004.

[Ike85] T Ikemura. Codon usage and tRNA content in unicellular and multicellular organisms. *Mol Biol Evol*, 2(1) :13–34, 1985.

[JDH00] T.S. Jaakkola, M. Diakhans, and D. Haussler. A discriminative framework for detecting protein remote homologies. *Journal of Computational Biology*, 7 :95–114, 2000.

[JH99] T.S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In S.A. Solla M.S. Kearns and editors D.A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1999.

[KAH+02] Anuj Kumar, Seema Agarwal, John A. Heyman, Sandra Matson, Matthew Heidtman, Stacy Piccirillo, Lara Umansky, Amar Drawid, Ronald Jansen, Yang Liu, Kei-Hoi Cheung, Perry Miller, Mark Gerstein, G. Shirleen Roeder, and Michael Snyder. Subcellular localization of the yeast proteome. *Genes Dev.*, 16(6) :707–719, 2002.

[KCC03] Yu-Dong Cai Kuo-Chen Chou. Prediction and classification of protein subcellular location ?- ?sequence-order effect and pseudo amino acid composition. *Journal of Cellular Biochemistry*, 90 :1250–1260, 2003.

[KH66] S. Karlin and M. Taylor Howard. *A first course in stochastic processes*. New York : Academic Press, 1966.

[KKS04] Lukas Käll, Anders Krogh, and Erik L. L. Sonnhammer. A combined transmembrane topology and signal peptide prediction method. *Journal of Molecular Biology*, 338 :1027–1036, May 2004.

[LCB+04] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5 :27–72, 2004.

[LEC+04] Christina S. Leslie, Eleazar Eskin, Adiel Cohen, Jason Weston, and William Stafford Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4) :467–476, 2004.

[LSG+04] Z. Lu, D. Szafron, R. Greiner, P. Lu, D.S. Wishart, B. Poulin, J. Anvik, C. Macdonell, and R. Eisner. Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, 20(4) :547–556, 2004.

[MAPM04] Jakob Christian Mueller, Christophe Andreoli, Holger Prokisch, and Thomas Meitinger. Mechanisms for multiple intracellular localization of human mitochondrial proteins. *Mitochondrion*, 3(6) :315–325, May 2004.

[McG85] Duncan J. McGeoch. On the predictive recognition of signal peptide sequences. *Virus Research*, 3 :271–286, October 1985.

[MS00] L. Marsan and M.-F. Sagot. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *J. Computational Biology*, 7 :345–362, 2000.

[MSBP02] Richard Mott, Jorg Schultz, Peer Bork, and Chris P. Ponting. Predicting Protein Cellular Localization Using a Domain Projection Method. *Genome Res.*, 12(8) :1168–1174, 2002.

[Mur97] Florence Muri. *Comparaison d'algorithmes d'identification de chaînes de Markov cachées et application à la détection de régions homogènes dans les séquences d'ADN*. PhD thesis, Université René Descartes, Paris V, october 1997.

[NBM+02] Pierre Nicolas, Laurent Bize, Florence Muri, Mark Hoebeke, Francois Rodolphe, S. Dusko Ehrlich, Bernard Prum, and Philippe Bessieres. Mining Bacillus subtilis chromosome heterogeneities using hidden Markov models. *Nucl. Acids. Res.*, 30(6) :1418–1426, 2002.

[NBM+05] Mitsuteru Nakao, Roberto A. Barrero, Yuri Mukai, Chie Motono, Makiko Suwa, and Kenta Nakai. Large-scale analysis of human alternative protein isoforms : pattern classification and correlation with subcellular localization signals. *Nucl. Acids Res.*, 33(8) :2355–2363, 2005.

[NCR03] Rajesh Nair, Phil Carter, and Burkhard Rost. NLSdb : database of nuclear localization signals. *Nucl. Acids Res.*, 31(1) :397–399, 2003.

[NEBvH97] Henrik Nielsen, Jacob Engelbrecht, Søren Brunak, and Gunnar von Heijne. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering*, 10 :1–6, 1997.

[NH99] Kenta Nakai and Paul Horton. Psort : a program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends in Biochemical Sciences*, 24 :34–35, January 1999.

[NK92] K. Nakai and M. Kanehisa. A knowledge base for predicting protein localization sites in eucaryotic cells. *Genomics*, 14(4) :897–911, 1992.

[NMSE+03a] Georg Neuberger, Sebastian Maurer-Stroh, Birgit Eisenhaber, Andreas Hartig, and Frank Eisenhaber. Motif refinement of the peroxisomal targeting signal 1 and evaluation of taxon-specific differences. *Journal of Molecular Biology*, 328 :567–579, May 2003.

[NMSE+03b] Georg Neuberger, Sebastian Maurer-Stroh, Birgit Eisenhaber, Andreas Hartig, and Frank Eisenhaber. Prediction of peroxisomal targeting signal 1 containing proteins from amino acid sequence. *Journal of Molecular Biology*, 328 :581–592, May 2003.

[NN92] H Nakashima and K Nishikawa. The amino acid composition is different between the cytoplasmic and extracellular sides in membrane proteins. *FEBS Letter*, 303 :141–146, 1992.

[NR02a] Rajesh Nair and Burkhard Rost. Inferring sub-cellular localization through automated lexical analysis. *Bioinformatics*, 18(90001) :78S–86, 2002.

[NR02b] Rajesh Nair and Burkhard Rost. Sequence conserved for subcellular localization. *Protein Sci*, 11(12) :2836–2847, 2002.

[NR03] Rajesh Nair and Burkhard Rost. Loc3d : annotate sub-cellular localization for protein structures. *Nucl. Acids Res.*, 31(13) :3337–3340, 2003.

[NR05] Rajesh Nair and Burkhard Rost. Mimicking cellular sorting improves prediction of subcellular localization. *Journal of Molecular Biology*, 348 :85–100, April 2005.

[Nue] G. Nuel. Numerical solutions for pattern statistics on markov chains. soumis à Journal of Computational Biology.

[OS98] Matej Oresic and David Shalloway. Specific correlations between relative synonymous codon usage and protein secondary structure. *Journal of Molecular Biology*, 1998.

[OSW05] C.S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6 :1043–1071, 2005.

[PCST00] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin dags for multiclass classification. In S.A. Solla, T.K. Leen, and K.-R. Mueller, editors, *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.

[Pet69] T. Petrie. Probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 40(1) :97–115, 1969.

[PK03] Keun-Joon Park and Minoru Kanehisa. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19(13) :1656–1663, 2003.

[PL88] WR Pearson and DJ Lipman. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci.*, 85(8) :2444–2448, 1988.

[Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceeding of the IEEE*, 77 :257–286, 1989.

[RD99] S. Robin and J.J. Daudin. Exact distribution of word occurrences in a random sequence of letters. *J. Appl. Prob.*, 36 :179–193, 1999.

[RD00] S. Robin and J.J. Daudin. Exact distribution of the distances between any occurrences of a set of words. *Ann. Inst. Statist. Math.*, 2000.

[RH98] A Reinhardt and T Hubbard. Using neural networks for prediction of the subcellular location of proteins. *Nucl. Acids. Res.*, 26(9) :2230–2236, 1998.

[RSW00] G. Reinert, S. Schbath, and M. Watermann. Probabilistic and statistical properties of words : An overview. *Journal of Computational Biology*, 7 :1–46, 2000.

[SCLK05] Deepak Sarda, Gek Chua, Kuo-Bin Li, and Arun Krishnan. pslip : Svm based protein subcellular localization prediction using multiple physicochemical properties. *BMC Bioinformatics*, 6(1) :152, 2005.

[SL87] PM Sharp and WH Li. The codon Adaptation Index–a measure of directional synonymous codon usage bias, and its potential applications. *Nucl. Acids Res.*, 15(3) :1281–1295, 1987.

[SLHT04] M. Scott, G. Lu, M. Hallett, and D. Y. Thomas. The Hera database and its use in the characterization of endoplasmic reticulum proteins. *Bioinformatics*, 20(6) :937–944, 2004.

[SS01] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[STH04] Michelle S. Scott, David Y. Thomas, and Michael T. Hallett. Predicting Subcellular Localization via Protein Motif Co-Occurrence. *Genome Res.*, 14(10a) :1957–1966, 2004.

[TKR+02] K Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.R. Müller. A new discriminative kernel from probabilistic models. In G. Dietterich, S. Becker, and Z. editors Ghahramani, editors, *Advances in Neural Information Processing Systems*. MIT Press, 2002.

[Vap95] V. Vapnik. *The nature of Statistical Learning Theory*. Springer, New-York, 1995.

[Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley, New-York, 1998.

[Ver02] J.P. Vert. Support vector machine prediction of signal peptide cleavage site using a new class of kernels for strings. In *Pacific Symposium Biocomputing*, pages 649–660, 2002.

[vH86] G. von Heijne. A new method for predicting signal sequence cleavage sites. *Nucleic Acids Research*, 14 :4683–4690, June 1986.

[vHACV98] J. van Helden, B. André, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computationnal analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281 :827–842, 1998.

[vHdOPO00] Jacques van Helden, Marcel.li del Olmo, and Jose E. Perez-Ortin. Statistical analysis of yeast genomic downstream sequences reveals putative polyadenylation signals. *Nucl. Acids. Res.*, 28(4) :1000–1010, 2000.

[VHRCV00] J. Van Helden, A. F. Rios, and J. Collado-Vides. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Research*, 28(8) :1808–1818, 2000.

[VMS99] A. Vanet, L. Marsan, and M.-F. Sagot. Promoter sequences and algorithmical methods for identifying them. *Res. Microbiol.*, 150 :779–799, 1999.

[WW99] J. Weston and C. Watkins. Multi-class support vector machines. In *Proceedings ESANN*, 1999.

[YDC02] Xiao-Jun Liu Xue-biao Xu Kuo-Chen Chou Yu-Dong Cai. Support vector machines for prediction of protein subcellular location by incorporating quasi-sequence-order effect. *Journal of Cellular Biochemistry*, 84 :343–348, 2002.

[Yua99] Z Yuan. Prediction of protein subcellular locations using markov chain models. *FEBS letter*, 451 :23–26, 1999.

Prédiction de la localisation cellulaire des protéines par leurs séquences biologiques.

Les compartiments cellulaires, de par les frontières membranaires qui les définissent, permettent l'accomplissement de taches métaboliques diverses au sein de la cellule. Cette spécialisation en domaines intracellulaires induit donc une différentiation dans la fonction des protéines qui les composent. Le grand nombre de gènes orphelins produits ces dernières années par les projets de séquençage motive la mise au point de méthodes efficaces pour la prédiction ab-initio de la localisation cellulaire des protéines.
Ainsi la majorité de ce travail de thèse s'intéresse au problème de la prédiction du compartiment cellulaire d'une protéine à partir de sa séquence primaire.
Nous nous sommes attachés à proposer des alternatives descriptives aux méthodes existantes de prédiction de la localisation cellulaire en utilisant : (1) de nouveaux descripteurs issus de la séquence nucléique, (2) une approche par chaînes de Markov cachées (CMC) et arbres de décision. L'approche par CMC est justifiée biologiquement a posteriori car elle permet la modélisation de signaux d'adressage conjointement à la prise en compte de la composition globale. En outre, l'étape de classification hiérarchique par arbre améliore nettement les résultats de classification. Les résultats obtenues lors des comparaisons avec les méthodes existantes et utilisant des descripteurs fondés sur la composition globale possèdent des performances similaires.

Mots-clés : prédiction de la localisation cellulaire des protéines, chaînes de Markov, chaînes de Markov cachées, Support Vector Machines.

Predicting proteins subcellular localization by their biological sequences.

Cellular compartments, due to the membrane frontiers they induce, allow the realization of diverse metabolic tasks in the cell. This specialization of the cell's spatial domains directly corresponds to a differentiation in the functional role of the proteins they involve. Thus, when homology searches in the databanks produce no results, the knowledge of the localization site of a protein can help in deducing its function. With the large amount of unannotated orphan genes produced these last years, ab-initio prediction of the subcellular location of proteins has become an important problem.
Thus the major part of the work presented here concerns the prediction of the subcellular localization of a protein, knowing its primary or coding sequence.
We proposed descriptive alternatives to existing methods for predicting subcellular localization by : (1) using new descriptors from nucleotidic sequence and (2) an HMM approach combined with decision trees. The HMM approach is justified biologically in that it permits to modelize biological adressing signals conjointly with global composition. Furthermore, embedding the classification steps within a decision tree slightly improved classification results, whose accuracy is similar with other methods using global composition information.

Keywords : Markov chain, Hidden Markov Models, Support Vector Machines, protein subcellular localization prediction.