Université d'Évry-Val d'Essonne
U.F.R. Sciences Fondamentales et Appliquées

THÈSE

présentée pour obtenir

le grade de Docteur en Sciences
de l'Université d'Évry-Val d'Essonne

Spécialité
bioinformatique, biologie structurale et génomique

Cédric Auliac

Approches évolutionnaires pour la reconstruction de réseaux de régulation génétique par apprentissage de réseaux Bayésiens.

Soutenue le *24 Septembre 2008* devant le jury composé de

M. Pascal Barbry	*Président du jury*
M. Philippe Leray	*Rapporteur du jury*
M. Louis Wehenkel	*Rapporteur du jury*
M. José Peña	*Examinateur*
M. Vincent Frouin	*Encadrant CEA*
M^{me} Florence d'Alché-Buc	*Directrice de thèse*

Thèse préparée au Laboratoire d'Exploration Fonctionnelle des Génomes
Service de génomique Fonctionnelle
Commissariat à l'Énergie Atomique (CEA)

Ainsi que dans l'équipe AmisBio du laboratoire IBISC
FRE 2873 CNRS / Université d'Évry-Val d'Essonne

à mes parents

à françois

Remerciements

Tout d'abord je remercie Florence et Vincent pour m'avoir guidé durant ces quatre années. Effectivement, vous n'étiez pas trop de deux pour venir à bout du Auliac, et je ne m'en plains pas ! Aussi compliqué que soit le travail à trois, j'espère être parvenu à tenir autant de l'universitaire que de l'ingénieur. Vous suivre était certainement la meilleur formation que je puisse espérer.

Je tiens à remercier les membres du jury qui m'ont fait l'honneur de m'accompagner dans la phase finale de cette thèse, à commencer par mes rapporteurs, Philippe Leray et Louis Wehenkel. Leurs commentaires et leurs questions m'ont permis de prendre un recul salutaire par rapport à mes travaux. Je remercie très chaleureusement José Peña pour avoir accepté de siéger dans mon jury, ainsi que Pascal Barbry pour avoir accepté de le présider. Je me sens particulièrement honoré de l'intérêt que ces deux spécialistes de disciplines pourtant fort différentes ont manifesté pour mes travaux.

Un grand merci à Xavier Gidrol pour m'avoir ouvert les portes de son laboratoire, pour m'avoir financé et surtout pour m'avoir accordé sa confiance. Merci également à Gilles Bernot pour m'avoir permis de faire mes premières armes en informatique.

Je remercie évidemment la jeunesse atomique, à commencer par les membres de l'équipe bio-info : Cyril et Olivier pour l'accueil qu'ils m'ont fait dès mon arrivée en DEA, ainsi que Cathy, Junior et Arthur pour leurs coups de main salvateurs dans les coups de bourre.

Merci à tous ceux grâce auxquels je me suis senti chez moi au SGF durant ces quatre années : Amandine, Peggy, la douce Amélie, Jérôme, Valérie, Ghida, le couple suricate, les partenaires de la puce, Johnny, Arthur, Chtulu ainsi que les grands anciens, Yoann, Sandrine, Gaétan, Alex, Thierry, Calou, Florian, Walid, Morad, René le bucheron, truffana et la truffe du piémont (la différence est subtile mais réelle).

Une dédicace particulière à la brigade des coquins : Natacha et sa précieuse contribution au maintien d'un équilibre de vie fondé sur le cognac et l'optimisme ; la délicieuse Elisabeth dont les talents de couturière ainsi que l'amour des bonnes liqueurs sont une source d'inspiration quotidienne ; et Simon, le jeune et brillant apprenti parti apporter la bonne parole sous les cieux du privé.

A poussin, mon jumeau maléfique d'obédience pharmacienne, pour avoir partagé avec moi ces années de thèse dans ce qu'elles ont de meilleur et de pire, en suivant toujours la même philosophie : manger gros, déguster un bon vin, faire un somme, puis enfiler son Bogdanov et repartir au quart de tour parce que c'est pas tout ça mais quand on ne sait pas où on va, et bien il faut y aller... et le plus vite possible ! Merci poussin d'y avoir couru à mes côtés.

Comme je suis un homme chanceux j'ai également été adopté par les gens d'IBISC qui ont su démontrer avec brio que l'on peut être informaticien ET drôle ! En premier lieu, mes remerciements vont aux membres de mon équipe : Pierre, Farida, et Nicolas dont l'énergie débordante et la gentillesse furent une réelle source de motivation. Mes pensées vont également aux AmisBoys avec qui j'ai partagé cette aventure : Cyril, Minh, et Nizar qui fut le seul à s'être montré digne de me ravir le titre d'homme le plus bavard de l'équipe. Avec qui pourrai-je parler cuisine, bandes dessinées et cinéma sans discontinuer à présent ? Au petit dernier, Julien, je souhaite de perpétuer cette prestigieuse lignée.

Je n'oublie pas les adeptes du petit bain avec qui j'ai pris du poids dans la bonne humeur : François et Stéphane à qui je souhaite tout le bonheur du monde, Popo mon co-bureau mythique, Thomas, Amandine, Sylvie et ses Henris, Antoine (qui râle presque aussi bien que moi !), Assia et sa voix mélodieuse sans laquelle les bureaux semblent vides, et Matthieu dont les fiches classées par ordre alphabétique des couleurs recèlent tous les secrets de la création, hormis ce qu'il pense vraiment et 42... Enfin, merci à Jean-Louis pour son attention et ses précieux conseils.

Un grand merci à ma mère, à mon père et à Josiane pour leur indéfectible soutien, je leur doit à peu près tout et plus encore.

Merci à ma famille pour sa bienveillance et sa patience face à des études un peu longues !

A biquet, pour nos virées nocturnes, nos discussions enflammées et les mauvais films qui font retomber la pression. Surtout, merci de raconter à tout le monde que je suis Will Hunting même si tu es persuadé qu'en réalité, je change des pneus dans une station service pour vivre ;-).

Merci à Ambre pour tout ce qu'elle a partagé avec moi, tout ce qu'elle m'a fait découvrir. Je lui souhaite toute la réussite et le bonheur qu'elle mérite.

Aux 4 pour leur amitié et pour m'avoir si souvent botté les fesses pour me faire avancer. Si il est vrai qu'un ami doit savoir frapper là où ça fait mal quand c'est pour votre bien, alors il n'en est pas de meilleurs qu'Alain et Stéphane. Vous avez ouvert la voie, à présent que l'école est fini, je vais m'efforcer de suivre votre exemple. Une pensée pour le Matthieu-garrou, coincé au pays des Caribous. Enfin, merci à Fabrice, moins pour son soutien passé que pour celui qu'il m'accordera toujours. Après tout, Tokyo ce n'est pas si loin, n'est-ce pas beudaille ?

Enfin, merci à toi, mon ange, pour ton aide, ton soutien et ta présence. Grâce à toi, je suis probablement l'une des rares personnes à pouvoir prétendre avoir eu une dernière année de thèse heureuse.

merci

Sommaire

Liste des figures

Liste des tableaux

Introduction

De nos jours, la biologie s'inscrit dans une perspective systémique selon laquelle les cellules, qui sont les composantes de base du vivant, sont régies par le fonctionnement coordonné de multiples entités : gènes, protéines et métabolites. L'ensemble des interactions concertées entre ces entités constituent des réseaux de régulation biologique dont l'élucidation est l'un des objectifs majeurs de la biologie des systèmes. Elle repose essentiellement sur la mise en évidence et la caractérisation à un niveau global des relations entre ces entités.

Parmi les différents mécanismes de régulation à l'œuvre dans la cellule, il est communément admis que la régulation transcriptionnelle joue un rôle prépondérant. Ce processus de régulation génétique est d'autant plus important qu'il est pour l'instant le plus aisément observable, du fait notamment de la disponibilité de techniques expérimentales adaptées telles que les puces à ADN. Les données qui en sont issues rendent compte de l'activité transcriptionnelle de l'ensemble des gènes d'un échantillon de cellules dans des conditions expérimentales spécifiques. L'exploitation de ces données doit permettre l'extraction de connaissances en vue d'améliorer la compréhension de certains processus normaux ou pathologiques et de cerner de nouvelles cibles thérapeutiques. Ces travaux de recherche concernent l'apprentissage automatique des réseaux de régulation transcriptionnelle, à partir de données de transcriptome.

Cette tâche est généralement entreprise de la manière suivante. Dans un premier temps, une classe de modèles mathématiques permettant de décrire les interactions entre des gènes régulateurs et leurs gènes cibles est choisie. Les données sont ensuite utilisées afin d'apprendre à la fois le graphe d'interaction et les paramètres du modèle représentant le réseau de régulation. Nous considérons ici le cas où la structure même du modèle, c'est-à-dire le graphe d'interaction, est inconnue. La tâche d'apprentissage consiste donc à découvrir la nature des interactions entre gènes.

Les réseaux bayésiens. Jusqu'à présent, de nombreux formalismes mathématiques ont été utilisés pour modéliser les réseaux de régulation et étudier leur dynamique. Cependant, depuis quelques années, la plupart des modèles utilisés pour l'inférence de réseaux appartiennent à la famille des modèles graphiques probabilistes. Les réseaux bayésiens, plus particulièrement, offrent un cadre intéressant pour la représentation de dépendances entre gènes. Ils permettent

notamment de modéliser le caractère stochastique de la mécanique du vivant et d'intégrer l'incertitude inhérente aux observations tout en demeurant aisément interprétables.

Apprentissage de structure dans les réseaux bayésiens. L'apprentissage de réseaux bayésiens est généralement considéré à travers un problème de sélection de modèle. On se donne un critère de performance permettant d'évaluer la qualité d'un modèle candidat. Il est généralement fondé sur la vraisemblance du modèle, c'est-à-dire sur sa capacité à expliquer les données. Il est courant d'y inclure une contrainte favorisant les modèles les plus simples, afin de lutter contre le sur-apprentissage. Une méthode d'exploration doit permettre d'identifier parmi tous les modèles possibles celui qui optimise ce critère. Toutefois, l'exploration de l'espace des modèles possibles est NP-difficile. Pour résoudre ce problème d'optimisation combinatoire nous avons recours aux algorithmes évolutionnaires. Le principe de cette approche consiste à faire évoluer artificiellement un ensemble de modèles candidats au moyen de deux processus complémentaires : un processus de recombinaison visant à échanger les caractéristiques des modèles entre eux et un processus de sélection qui permet de ne retenir au fil des générations que les modèles les plus simples expliquant le mieux les données, c'est à dire, les modèles qui optimisent le critère de performance.

Approches évolutionnaires pour l'inférence de structure. De nombreuses stratégies de reproduction et de sélection étant envisageables, nous menons dans un premier temps une étude comparative afin d'identifier les plus performantes. Nous mettons notamment en évidence le rôle fondamental des mécanismes de spéciation. Ces derniers, en permettant d'entretenir une population de solutions hétérogènes, évitent de converger prématurément vers une population homogène de solutions sous-optimales. Les méthodes ainsi retenues sont ensuite utilisées pour valider l'approche évolutionnaire par comparaison avec des méthodes d'apprentissage préexistantes.
Pour mener à bien cette étude, il est important de pouvoir contrôler la quantité ainsi que la qualité des données employées pour l'apprentissage. Il est également nécessaire d'avoir à sa disposition le modèle étant à l'origine des données, pour être en mesure de juger de la qualité des modèles appris. C'est pourquoi nous utilisons pour nos expériences des données d'expression synthétiques, obtenues par échantillonnage d'un réseau artificiel bio-réaliste précédemment utilisé dans la littérature [LBU04] et modélisant l'homéostasie du glucose : le réseau Insuline. Afin de nous placer dans des conditions réalistes, nous générons des jeux de données de taille restreinte, comptant typiquement quelques centaines de données.
Nous achevons ces travaux par une étude propective, portant sur l'utilisation des algorithmes à estimation de distribution (EDA) pour l'apprentissage de structure de réseaux bayésiens. Nous montrons que dans le cadre que nous nous sommes fixé, cette famille d'algorithmes constitue une alternative prometteuse aux algorithmes évolutionnaires préalablement testés.

Organisation du document

Le document se structure en 3 parties.

Dans la première partie, « Introduction à l'étude systémique des fonctions cellulaires », nous présentons l'objet de notre étude, à savoir les réseaux de régulation génétique, à travers la problématique de la biologie des systèmes : comprendre le comportement collectif des molécules biologiques et leurs interactions au sein de la cellule.

- Le chapitre 1 (page 9) présente de manière plus fine le contexte scientifique de la biologie des systèmes. Nous développons la thématique des réseaux de régulation biologique en apportant un éclairage plus particulier sur les réseaux de régulation génétique.
- Dans le chapitre 2 (page 21), nous présentons les principales techniques expérimentales permettant d'étudier le comportement global des gènes dans un tissu, sous certaines conditions expérimentales. Nous détaillons le principe des puces à ADN ainsi que les principaux aspects techniques de leur mise en œuvre. Nous décrivons ensuite les différentes étapes de traitement des données brutes des puces à ADN visant à les rendre exploitables. Enfin, nous évoquons certaines techniques permettant d'interpréter les données traitées et d'en extraire de l'information.

Dans la seconde partie, « Apprentissage des réseaux de régulation génétique », nous présentons la problématique d'apprentissage de réseaux de régulation à travers l'étude de différents formalismes mathématiques utilisés pour la modélisation des réseaux de régulation et l'étude de leur dynamique.

- Le chapitre 3 (page 47) présente dans un premier temps différents formalismes essentiellement dédiés à l'étude dynamique des réseaux de régulation. Nous discutons les opportunités d'apprentissage de ces modèles à partir de données de puces à ADN. Dans un second temps, les modèles à base de graphe sont abordés et leur apprentissage au moyen de méthodes statistiques éprouvées est détaillé. Nous finissons par la présentation des modèles graphiques Gaussiens, qui appartiennent à la famille des modèles graphiques probabilistes non orientés.
- Le chapitre 4 (page 77) est dédié au problème d'apprentissage des réseaux bayésiens et plus particulièrement de la structure de ces modèles probabilistes à partir de données. Nous commençons par introduire les modèles graphiques orientés et plus particulièrement les réseaux bayésiens qui nous semblent constituer un support adéquat pour l'apprentissage de réseaux de régulation génétique. Un tour d'horizon des différentes méthodes d'apprentissage de structure existantes est effectué.

Dans la troisième partie, « Apprentissage évolutionnaire des réseaux Bayésiens », nous présentons les approches évolutionnaires développées dans cette étude ainsi que les résultats numériques ayant permis d'éprouver leurs qualités respectives.

- Dans le chapitre 5 (page 115), nous insistons sur les outils que nous avons retenus pour entreprendre la sélection de modèle dans ces travaux et notamment sur les algorithmes évolutionnaires. Nous présentons certains principes généraux de cette classe d'algorithmes et nous mettons en avant les adaptations nécessaires à leur utilisation dans le cadre de l'apprentissage de structure de modèles graphiques orientés.
- Le chapitre 6 (page 133) détaille les expériences menées pour comparer les mérites des stratégies que nous avons développées dans ce cadre évolutionnaire. Nous insistons sur la mise en perspective du problème d'optimisation traité avec les problèmes biologiques précis que nous souhaitons aborder. Enfin, les résultats obtenus avec des algorithmes génétiques sont comparés à ceux produits par un ensemble de méthodes d'apprentissage célèbres.
- Le chapitre 7 (page 155) introduit la famille des algorithmes à estimation de distribution (EDA) et présente des travaux préliminaires sur l'application de cette classe de méthode à l'apprentissage de structure dans les réseaux bayésiens. Les résultats obtenus dans ce chapitre sont mis en perspective avec ceux obtenus plus tôt avec les algorithmes génétiques.

Enfin, la conclusion (page 173) récapitule les travaux effectués durant cette thèse avant d'en

présenter les perspectives.

Notons que les termes suivis d'une astérisque dans la suite de ce mémoire sont explicités dans le glossaire fourni à la page 179.

Première partie

Introduction à l'étude systémique des fonctions cellulaires

Chapitre 1

La cellule, un système d'interactions régulatrices

La biologie, au même titre que l'économie ou la météorologie, est un domaine où les approches de type systèmes complexes revêtent un intérêt fondamental. Il est en effet établi que le phénotype d'une cellule repose sur le fonctionnement coordonné des nombreuses entités qui la composent. Il est difficile d'en saisir le fonctionnement global à partir de l'étude spécifique de l'un de ses constituants. Une des définitions de la biologie des systèmes considère l'étude du fonctionnement global des cellules composant un organisme en considérant que chacune d'entre elles est un système complexe. De nombreux résultats sur les fonctions cellulaires ont été produits grâce à l'étude spécifique de certains gènes ou molécules. C'est par exemple le cas de l'élucidation de certaines voies métaboliques ou de voies de signalisation. On parle alors souvent de *biologie dédiée*, fondée sur l'étude d'un petit nombre de molécules, au moyen de techniques de biologie dites « mono-molécules ». Cette approche peut être opposée à la biologie intégrative (bien que dans les faits ces deux approches se nourrissent l'une l'autre) consistant en une approche plus globale visant à étudier un grand nombre de molécules biologiques en parallèle au moyen de techniques telle que les puces à ADN que nous présentons dans le chapitre suivant. L'étude de molécules clefs, connues pour jouer un rôle prépondérant dans certains processus cellulaires, est certes riche en enseignement et fournit des points d'entrée pour comprendre certains phénomènes cliniques ou biologiques. Toutefois cette approche qui ne considère qu'une seule molécule ou une seule réaction à la fois a ses limites. En effet, le comportement normal ou pathologique de la cellule tient plus à la dynamique des interactions entre les molécules ou à leur répartition tridimensionnelle qu'à des interactions isolées et indépendantes.

Les objectifs principaux de la biologie des systèmes sont à la fois d'ordre théorique et applicatif. Le premier objectif est, comme nous l'avons évoqué précédemment, une meilleure compréhension des mécanismes qui sous-tendent les grandes fonctions cellulaires : quels en sont les acteurs (gènes, protéines, métabolites...), comment ces derniers interagissent-ils et surtout comment une fonction peut-elle être modulée ou supprimée tant dans un contexte biologique que par des moyens artificiels. Il s'agit donc avant tout d'approfondir notre connaissance du vivant. La compréhension poussée du fonctionnement normal et anormal de la cellule doit permettre — dans une certaine mesure — de proposer des hypothèses sur des aspects fonctionnels de certaines pathologies et de proposer des stratégies thérapeutiques adaptées. Au-delà de l'aspect fondamental, ce sont donc les applications biomédicales qui sont visées.
On peut notamment citer le diagnostique et l'étude de prédispositions aux maladies fondés non plus seulement sur la présence ou l'absence des certains allèles constituant des facteurs de

risque mais aussi sur un mode de fonctionnement particulier des gènes concernés. Ce qui nous intéresse, c'est la façon dont ces gènes s'expriment pour produire des molécules (classiquement des protéines) fonctionnelles dans une concentration donnée. Pour que cela ait un sens, il importe de caractériser cette expression génétique conjointement pour un ensemble de gènes, ces derniers pouvant influer sur l'activité de leurs semblables dans le temps. En effet, de la même manière que la fluidité de la circulation sur une voie d'autoroute dépend moins des véhicules qui s'y trouvent que de leurs positions et de leurs vitesses relatives, c'est moins l'état individuel des gènes à un instant t que leurs influences mutuelles dans le temps et le comportement global qui en résulte qui nous intéressent. Pour poursuivre sur le parallèle avec l'étude du trafic autoroutier, le passage d'un état de fonctionnement normal (circulation fluide/cellule saine) à anormal (embouteillage/cellule malade) peut s'expliquer par le comportement collectif des véhicules qui l'empruntent et non pas seulement par la liste de ces derniers.
D'autres travaux portent sur l'étude des risques de nature exogène. Ils s'intéressent alors à la façon dont des facteurs pathogènes (virus ou bactéries) ou des toxiques (drogues, pesticides, rayonnements ionisants, etc.) perturbent les régulations entre des entités biologiques régissant certaines fonctions cellulaires, altérant ainsi le fonctionnement normal de la cellule.
De telles études doivent permettre l'apparition de stratégies thérapeutiques nouvelles, plus spécifiques et présentant des risques d'effets secondaires limités. En effet, la connaissance des interactions croisées entres les gènes permet de cibler spécifiquement ceux qui sont impliqués dans la pathologie au moyen de drogues ou de thérapies géniques, tout en évitant d'altérer d'autres fonctions vitales pour la cellule.

1.1 Les principaux acteurs moléculaires de la vie cellulaire

Le génome joue un rôle central dans le contrôle de processus cellulaires fondamentaux tels que la différentiation* , la division cellulaire* ou la réponse aux signaux environnementaux [1].

Tout d'abord, rappelons que le génome d'un organisme est la séquence génétique complète d'un ensemble de chromosomes, soit l'ensemble de l'information héréditaire encodée dans l'ADN. Cela inclut à la fois les gènes, qui sont des segments d'ADN codant chacun pour une molécule ayant une fonction biologique spécifique, et les séquences non codantes de l'ADN, qui sont dépourvues de l'information nécessaire à la synthèse de telles molécules. Par la suite nous considérerons le cas fréquent où les molécules codées par les gènes sont des protéines. Il faut cependant garder à l'esprit que les gènes peuvent également coder pour d'autres molécules telles que les ARN ribosomiques* (ARNr) ou encore des ARN particuliers qui participent à la régulation transcriptionnelle comme les micro-ARN* (miRNA) ou les petits ARN interférants* (siRNA).

Les protéines sont des macromolécules complexes pouvant remplir un très large éventail de fonctions. Les enzymes notamment sont des protéines pouvant catalyser des réactions biochimiques. Elles permettent de modifier de nombreux composés cellulaires et ce faisant de modifier leur comportement et leur action biologique. Plus généralement, les enzymes participent au métabolisme de la cellule, c'est-à-dire à l'ensemble des réactions permettant de décomposer les substances « ingérées » par la cellule et de synthétiser à partir de leurs « résidus » de nouvelles molécules exploitables (sources d'énergie, ou briques de bases de certaines molécules fonctionnelles telles que les protéines elles-mêmes). Différents types de protéines participent également à la transduction du signal cellulaire. Il s'agit du processus permettant à l'information environnementale perçue par la cellule au niveau de ces récepteurs membranaires d'être transmise et am-

[1] Comme nous le verrons par la suite, ces derniers peuvent être de natures diverses : les tissus d'un organisme et les cellules qui les composent sont susceptibles de réagir à des stimuli chimiques (drogues), physiques (rayonnements ionisants), mais aussi à des stress plus classiques tels que la fatigue, la faim et la maladie.

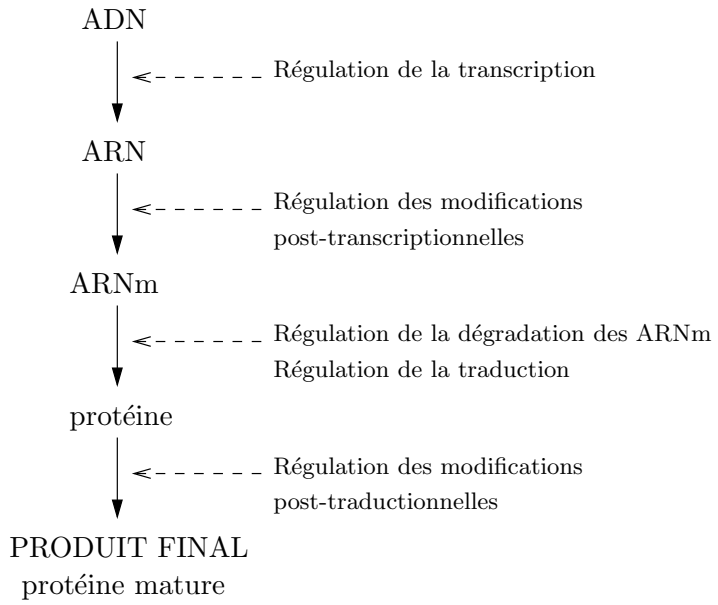


Fig. 1.1 – Schéma de la chaîne de biosynthèse des protéines.

plifiée au sein même de la cellule afin que celle-ci puisse s'adapter à un changement extérieur. Enfin, certaines protéines, les facteurs de transcription, peuvent se fixer dans les régions régulatrices de l'ADN afin de moduler l'activité transcriptionnelle des gènes.

1.2 Du gène à la protéine

On emploie fréquemment le mot « expression » pour désigner le fait que l'information portée par un gène est utilisée par la cellule pour synthétiser une protéine. On résume alors toute la chaîne de biosynthèse des protéines à cette seule notion. Comme on peut le voir sur la figure 1.1, la chaîne de biosynthèse d'une protéine est pourtant un processus complexe faisant intervenir de nombreuses étapes dont l'efficacité est susceptible d'être modulée par d'autres mécanismes cellulaires. On dit que ces étapes sont susceptibles d'être régulées.

Transcription La première étape — la mieux caractérisée — est la transcription. Il s'agit d'un mécanisme visant à produire un « négatif » de la séquence codante portée par un gène. Le mécanisme de transcription produit une molécule d'ARN dont la séquence est complémentaire de celle du gène transcrit, la différence majeure étant que les bases azotées constituant l'ARN sont différentes de celle de l'ADN et que l'ARN a une structure simple brin quand l'ADN a une structure en hélice double brin.

Modifications post-transcriptionnelles Cette molécule d'ARN subit ensuite diverses modifications visant à la rendre fonctionnelle et à faciliter son transport vers une région de la cellule ou l'information qu'elle porte pourra être exploitée. Parmi ces modifications post-transcriptionnelles figure l'épissage alternatif, qui consiste à découper une séquence d'ARN néosynthétisée afin d'en éliminer plusieurs segments superflus (appelés *introns*) avant de concaténer les segments restants (appelés *exons*) portant l'information génétique pertinente. La molécule obtenue au final est appelée ARN messager (ARNm). On notera qu'une même séquence d'ARN peut donner plusieurs ARNm distincts car les séquences éliminées

et conservées peuvent varier selon le mode d'épissage mis en œuvre ponctuellement par la cellule. De même une unique molécule d'ARN immature peut engendrer plusieurs ARNm matures à l'issue de la même phase d'épissage. Les remarques précédentes montrent que le dogme « 1 gène = 1 ARNm = 1 protéine » doit donc être fortement relativisé.

Traduction Chez les eucaryotes, les molécules d'ARNm permettent à l'information génétique codée dans les gènes de quitter le noyau pour être traitée dans le cytoplasme. Le processus de traduction y assure alors l'exploitation de l'information portée par les ARN messagers et permet de synthétiser à partir de ces derniers des protéines, qui sont les composants clefs de la vie cellulaire.

Modifications post-traductionnelles À leur tour les protéines néosynthétisées subissent des modifications post-traductionnelles afin de produire des protéines matures, pleinement fonctionnelles.

Nous n'allons pas détailler ici la nature des diverses réactions chimiques impliquées. Seul importe le fait que chacune des étapes mentionnées précédemment peut être régulée par des protéines codées par des gènes dont l'expression est elle-même susceptible d'être régulée par d'autres protéines.

1.3 Interactions moléculaires et influences régulatrices

Afin d'expliciter les phénomènes de régulations croisées que nous venons d'évoquer, il est utile de mettre en avant les différents types d'interactions existants entre les espèces moléculaires qui nous intéressent. Chacune d'entre elles sous-tend un mécanisme de régulation pouvant opérer à différents stades de la chaîne de biosynthèse des protéines.

Les interactions protéine-ADN Des protéines, appelées facteurs de transcription, peuvent se fixer en amont de la séquence codante d'un gène, au sein d'une séquence dite régulatrice, afin de contrôler son activité transcriptionnelle. L'enclenchement du mécanisme de transcription qui génère les ARNm est modulé par la présence d'un facteur de transcription sur son site cible.

Les interactions protéine-ARN(m) Comme nous l'avons déjà expliqué, les molécules d'ARN néosynthétisées doivent subir diverses modifications avant de constituer des ARNm matures. Toutes ces transformations reposent sur des complexes protéiques spécialisés. En jouant sur la composition ou la concentration des constituants de ces complexes, il est possible de réguler ce processus de maturation.
À leur tour, les ARN messagers sont traduits en protéines. Là encore, des protéines régulatrices peuvent moduler ce processus de traduction. Notons qu'il est également envisageable de favoriser la traduction en limitant la dégradation des ARNm qui sont des molécules très fragiles. Là encore, des protéines peuvent jouer un rôle stabilisateur pour les ARNm.

Les interactions protéines-protéines Les interactions physiques survenant entre des protéines peuvent recouvrir différents aspects. Certaines protéines sont synthétisées à l'état de monomères inactifs et doivent se combiner pour constituer une entité (polymère) fonctionnelle. C'est notamment le cas de nombreuses protéines impliquées dans la régulation transcriptionnelle qui n'agissent qu'une fois réunies au sein de complexes régulateurs. Par ailleurs, certaines enzymes interagissent avec d'autres protéines afin de les modifier chimiquement. Ces modifications permettent d'activer (ou au contraire d'inactiver) la fonction de la protéine cible. C'est notamment par ce type de processus que les protéines propagent

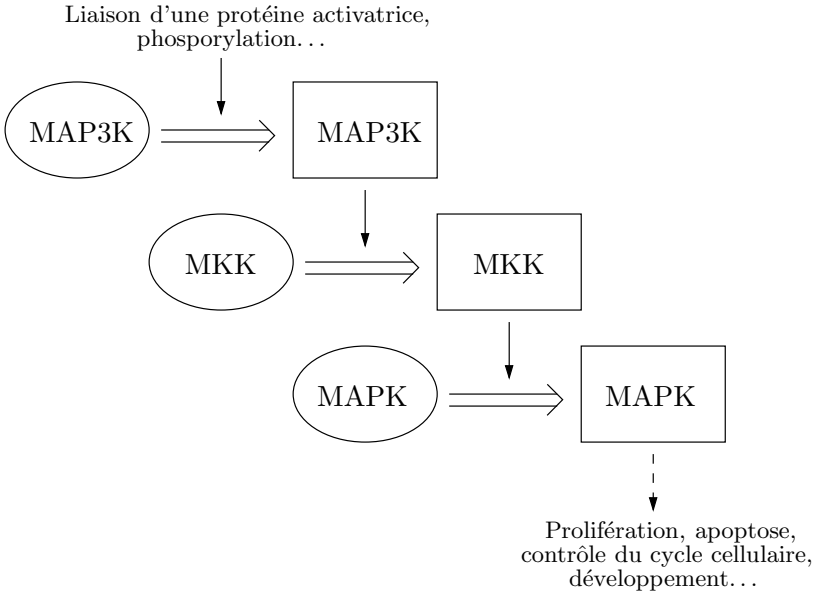


Fig. 1.2 – Une cascade d'activation de MAP-kinases. Les figures arrondies représentent les enzymes inactives, les figures rectangulaires représentent les même enzymes dans leur forme active. Les kinases mettent en oeuvre une fonction phosphorylante permettant d'activer une autre kinase. Par activations successives de kinases, un signal présenté à la membrane d'une cellule peut ainsi parvenir à l'intérieur du noyau et enclencher une fonction cellulaire.

un signal au sein de la cellule, par une cascade de modifications. La cascade des *MAP-kinases* est l'exemple classique de la propagation d'un signal par une suite d'interactions protéines-protéines. Un exemple est présenté à la figure 1.2. Une autre manière de réguler l'expression d'un gène consiste à modifier l'état de la chromatine, qui compose les chromosomes. Lorsqu'elle est « ouverte », elle permet à la machinerie transcriptionnelle d'accéder aux gènes, favorisant l'expression génique. Inversement, lorsqu'elle est « fermée » (c'est à dire très dense), la transcription des gènes est limitée. L'état de la chromatine est dicté par des modifications post-traductionnelles des protéines liées à l'ADN : les histones. Par exemple, l'acétylation (ajout de groupement acétyl) de ces protéines au niveau de résidus lysines entraîne le relâchement de la chromatine.

Les interactions protéines-métabolites Comme nous l'avons évoqué précédemment, la plupart des réactions chimiques survenant au sein de la cellule sont catalysées par des enzymes. Ces dernières doivent interagir avec leur substrat pour le modifier, qu'il s'agisse de sucres, de lipides, ou encore de protéines. Ces enzymes doivent être présentes en quantité suffisante et sous une forme fonctionnelle afin que le métabolisme cellulaire soit assuré.

1.4 Des réseaux de régulation biologique aux réseaux de régulation transcriptionnelle

Pour illustrer simplement l'enchevêtrement d'interactions et d'influences régulatrices évoquées précédemment, nous reprenons l'exemple fourni dans [dJ02].

Exemple 1.1
Un réseau de régulation élémentaire comportant 3 gènes codant pour des protéines inhibant

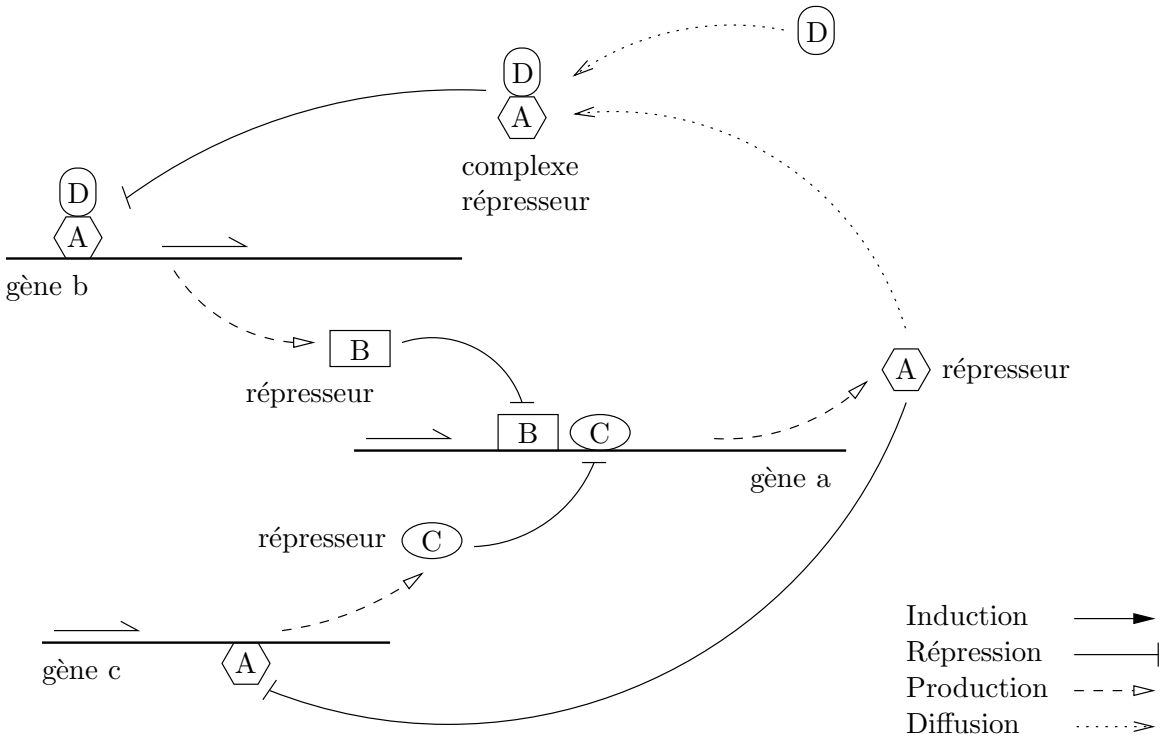


Fig. 1.3 – Un exemple de réseau de régulation biologique. Par convention, les gènes sont notés en minuscule alors que les noms des protéines (et autres molécules) sont notés en majuscule.

*l'expression d'autres gènes est présenté à la figure 1.3. Les répresseurs **B** et **C** se fixent sur différent sites de régulation de **a** (interactions protéines-ADN) pendant que **A** et **D** interagissent pour former un hétérodimère (interactions protéines-protéines) se fixant sur un site régulateur du gène **b**, empêchant l'ARN polymérase de transcrire la région en aval de ce site de fixation.*

Dans cet exemple, un ensemble de régulations biologiques structurées en réseau d'interaction entre ADN, ARN et protéines produit un système de régulation qui permet de contrôler des fonctions cellulaires. En fait, ce système de régulation peut également caractériser un type cellulaire. La seule identification des gènes, de leurs séquences régulatrices et des protéines susceptibles de venir s'y fixer apparaît insuffisante pour caractériser un type ou une fonction cellulaire. En effet, les cellules des différents types cellulaires partagent le même génome. Il est donc nécessaire de comprendre dans quelles conditions tels ou tels gènes sont exprimés, de quelle manière et dans quelle mesure.

Le séquençage du génome a permis la découverte d'un grand nombre de gènes et de sites de régulations. Dans la plupart des cas, les protéines impliquées dans la régulation de l'expression de ces gènes ainsi que les mécanismes mis en jeu ont été identifiés. Cependant, on en sait encore très peu quant à la façon dont ces phénomènes régulatoires fonctionnent de concert. Parvenir à comprendre comment des comportements cellulaires émergent des interactions entre gènes au sein de réseaux de régulation est donc un point crucial. Afin d'avancer dans cette compréhension, il est fondamental de disposer de techniques expérimentales permettant de disséquer les interactions et les influences régulatrices au niveau moléculaire. Il est aussi nécessaire de choisir un niveau du système de régulation constituant notre objet d'étude.

Les réseaux de régulation biologique permettent de conceptualiser un ensemble d'interactions biologiques sous la forme de réseaux, classiquement représentés par des graphes, dont les noeuds représentent les entités biologiques d'intérêt et les arcs les influences régulatrices entre ces entités. Ces réseaux sont généralement construits autour d'un seul type d'acteurs : les métabolites, les protéines ou les gènes. Ils se distinguent également par la nature des relations qui s'établissent entre ces acteurs : réactions chimiques entre métabolites, interactions physiques entre protéines et influences régulatrices abstraites entre gènes. Les trois grandes classes de réseaux biologiques rencontrés dans la littérature sont présentées ci dessous.

Les réseaux métaboliques Nous avons évoqué le métabolisme de la cellule qui lui permet de produire les composants dont elle a besoin à partir d'éléments extérieurs ou de les dégrader en fonction de ces besoins. Les réactions biochimiques sont elles-mêmes largement interdépendantes, les produits d'une réaction donnée pouvant servir de réactifs à d'autres réactions. Elles peuvent donc être étudiées de concert au moyen de réseaux métaboliques, deux métabolites étant liées s'il existe une réaction chimique (ainsi que l'enzyme permettant de la catalyser) permettant de produire l'une à partir de l'autre.

Les réseaux d'interactions protéine-protéine D'une manière générale, l'action coordonnée des protéines peut être décrite au moyen de réseaux d'interactions protéine-protéine. Les interactions en question sont des interactions physiques qui recouvrent essentiellement la formation de complexes protéiques (des monomères inactifs s'associent afin de former un polymère fonctionnel). Les réactions enzymatiques, lorsqu'elles portent sur des protéines et non sur des macromolécules telles que des sucres ou des acides gras, peuvent également être représentées dans un réseau d'interactions protéine-protéine. C'est par exemple le cas lorsqu'une enzyme **E** catalyse

une modification chimique d'une protéine **A** afin de bloquer (ou d'activer) sa capacité à se fixer sur une protéine **B** pour former un dimère **AB**.

Les réseaux de régulation génétique Ils représentent la façon dont chaque gène est susceptible de modifier (de réguler) le fonctionnement de l'un de ses semblables en modulant positivement (induction) ou négativement (répression) l'expression de ce dernier. Les gènes étant les plans de construction des ARN, ils ne peuvent interagir directement : dans les faits, l'expression d'un gène cible est régulée par le produit de son gène régulateur, une protéine dans la plupart des cas. Les réseaux de régulation génétique reposent donc sur des représentations plus abstraites que celles des réseaux d'interactions protéine-protéine ou des réseaux métaboliques : ils mettent en avant des liens de causalité générique entre gènes qui sont indépendants des mécanismes moléculaires précis mis en œuvre dans la cellule. Cette simplification est représentée dans la figure 1.4. La partie (1) de cette figure représente un réseau de régulation biologique « complet » alors que la partie (2a) représente le réseau de régulation génétique correspondant sous la forme d'un graphe orienté.

Les réseaux de régulation transcriptionnelle Bien que les interactions survenant au sein des différentes classes de macromolécules (ADN/ARN, protéines, métabolites) soient étroitement imbriquées, il est nécessaire de faire des hypothèses simplificatrices afin de réduire la complexité considérable du système étudié. En outre, sur le plan opérationnel, les compétences scientifiques et techniques requises pour étudier les ARN, les protéines et les métabolites sont très différentes et ne peuvent être menées en parallèle. L'étude des réseaux de régulation biologique intégrant toutes ces espèces moléculaires ne peut donc pas être envisagée en l'état. Le plus fréquemment, c'est le niveau transcriptionnel des réseaux de régulation biologique qui est retenu et étudié. Comme précisé précédemment, cela paraît réaliste sur le plan expérimental mais c'est également pertinent pour aborder tout une classe de problématiques fondamentales en biologie comme la différentiation, le développement ou la réaction à un toxique. Il s'agit d'étudier des fonctions cellulaires centrales pour lesquelles l'enjeu est de connaître les gènes qui influent sur l'activité transcriptionnelle de leurs semblables, et non sur leur capacité réelle à produire des protéines.

Ces influences régulatrices peuvent être représentées au sein de *réseaux de régulation transcriptionnelle (RRT)*. Ces derniers sont une simplification des réseaux de régulation génétique (RRG) dans la mesure où ils s'intéressent spécifiquement à la régulation de la quantité en ARNm des gènes et non à celle de leurs produits finaux qui, la plupart du temps, sont des protéines. C'est pourquoi par la suite, nous emploierons indifféremment les appellations RRG et RRT, ces deux concepts étant par ailleurs rarement distingués dans la littérature. Une justification théorique de cette simplification est que la transcription est une étape clef dans la régulation de la chaîne de biosynthèse des protéines, qui est particulièrement bien connue. Ayant fait le choix de ne considérer que la seule régulation transcriptionnelle au détriment des autres niveaux d'organisation, ce que nous considérons (des influences abstraites entre concentrations en ARNm de plusieurs gènes) est donc la superposition des réseaux d'interactions génétiques, protéiques et métaboliques.

Les réseaux de régulation transcriptionnelle revisités par les puces à ADN L'utilisation des RRT se fonde également sur des considérations d'ordre pratique et technique, les méthodes expérimentales d'étude du transcriptome étant particulièrement avancées. La technologie la plus couramment utilisée (que nous présentons dans la section suivante) est celle des puces à ADN qui permet de mesurer simultanément la concentration en ARN à l'équilibre d'un

ensemble de gènes. L'utilisation de cette source de données dans l'étude des RRT présente plusieurs inconvénients. D'abord, au sein d'un RRT, la concentration à l'équilibre d'un ARNm ne reflète pas nécessairement l'activité du gène qui l'a codé. Si cette concentration augmente, il est impossible d'attribuer cette variation à une induction de la transcription provoquant une augmentation de la production d'ARN (interaction protéine-ADN) ou à l'activation d'un processus de stabilisation de l'ARNm limitant sa dégradation (interaction protéine-ARN). Par ailleurs, les puces à ADN ne permettent pas de déterminer la nature de l'interaction moléculaire qui sous-tend une régulation transcriptionnelle (interaction protéine-ADN/ARN ou protéine-protéine). Ces problèmes ne sont cependant pas rédhibitoires dans la mesure où ce sont moins les détails des mécanismes moléculaires qui nous intéressent que leur finalité : l'augmentation ou la diminution de la quantité d'ARNm disponible au sein de la cellule. Dans le même ordre d'idée, les RRT sont censés représenter la façon dont des gènes modulent l'activité transcriptionnelle de leurs « voisins » par l'intermédiaire de leurs produits. Toutefois, il apparaît clairement que dans ce cadre et avec les outils d'investigation que nous nous sommes fixés, il est impossible de mesurer les effets éventuels d'une régulation traductionnelle ou post-traductionnelle.

Exemple 1.2
*On suppose qu'un gène **a** induit la traduction de **b**, tandis que **b** induit la transcription de **c**. Ce réseau de régulation est détaillé dans la partie (1) de la figure 1.4. Lorsque la quantité en ARNm de **a** augmente, celle de **b** demeure stable alors que celle de **c** augmente à son tour. Dans un RRT ces informations se traduisent par le fait que **a** induit **c** sans que **b** intervienne. Le réseau de régulation transcriptionnelle correspondant est représenté sur la partie (2b) de la figure 1.4.*

De nouveau, l'information tronquée dont on dispose reste pertinente car elle permet de capturer les relations, même indirectes, entre des acteurs d'un système de régulation gouvernant un phénomène biologique. Toutefois, si une modification de l'étape intermédiaire (un blocage de ***b*** par exemple) intervient sans que l'on puisse l'observer, il devient difficile d'expliquer la perte d'influence de ***a*** sur ***c***.
Enfin, le dernier inconvénient d'une approche uniquement centrée sur l'étude du transcriptome est que les influences régulatrices ne reposent pas nécessairement sur des interactions physiques directes. C'est pourquoi plusieurs scénarii peuvent expliquer une régulation transcriptionnelle.

Exemple 1.3
*Considérons l'exemple de la figure 1.5. Le réseau de régulation transcriptionnelle représenté sur la droite de la figure nous dit qu'un gène **a** inhibe un gène **d**. Les deux scénarii suivants (représentés dans la partie gauche de la figure) peuvent expliquer cette régulation.*

Influence directe *La protéine synthétisée par le gène **a** va se fixer sur la séquence promotrice du gène **d** afin d'inhiber sa transcription.*

Influence indirecte *La protéine synthétisée par le gène **a** interagit avec une protéine **B** afin de l'activer pour que celle-ci puisse à son tour capturer une protéine **C** et la retenir dans le cytoplasme. La protéine **C** étant un facteur de transcription du gène **d**, ce dernier ne peut être transcrit puisque l'un de ses facteurs de transcription ne peut pénétrer dans le noyau de la cellule.*

*En l'absence d'informations ou de données complémentaires, il est impossible de trancher entre ces deux hypothèses car dans les deux cas, les concentrations en ARNm des gènes **a** et **d** sont les seules susceptibles de varier : une augmentation de la concentration en ARNm de **a** provoque une diminution de celle de **d**. Notons qu'il est également envisageable qu'influences régulatrices directes et indirectes coexistent. Par exemple, il se peut que le produit d'un gène régulateur induise la transcription d'un gène cible à la fois de manière directe et de manière indirecte, par*

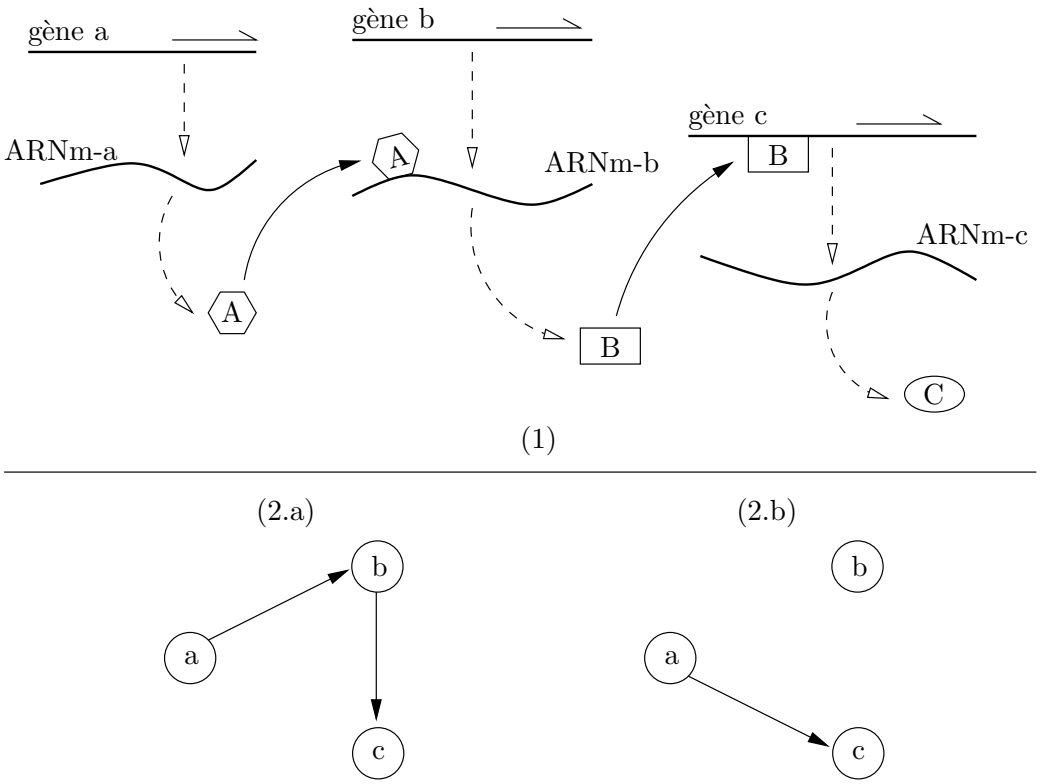


Fig. 1.4 – Des réseaux de régulation biologique aux réseaux de régulation génétique et transcriptionnelle. La figure (1) représente un réseau de régulation biologique où figurent différentes espèces moléculaires impliquées dans l'expression des gènes **a**, **b** et **c** : ADN, ARNm et protéines. Deux mécanismes de régulation interviennent : **A** régule la synthèse de la protéine **B** et **B** régule la synthèse de l'ARNm de **c**. Dans la sous-figure (2.a), le réseau de régulation génétique correspondant représente des régulations abstraites (indépendamment de leur nature) entre **a**, **b** et **c**. Dans la sous-figure (2.b), le réseau de régulation transcriptionnelle correspondant ne représente que la régulation transcriptionnelle indirecte de **a** sur **c**.

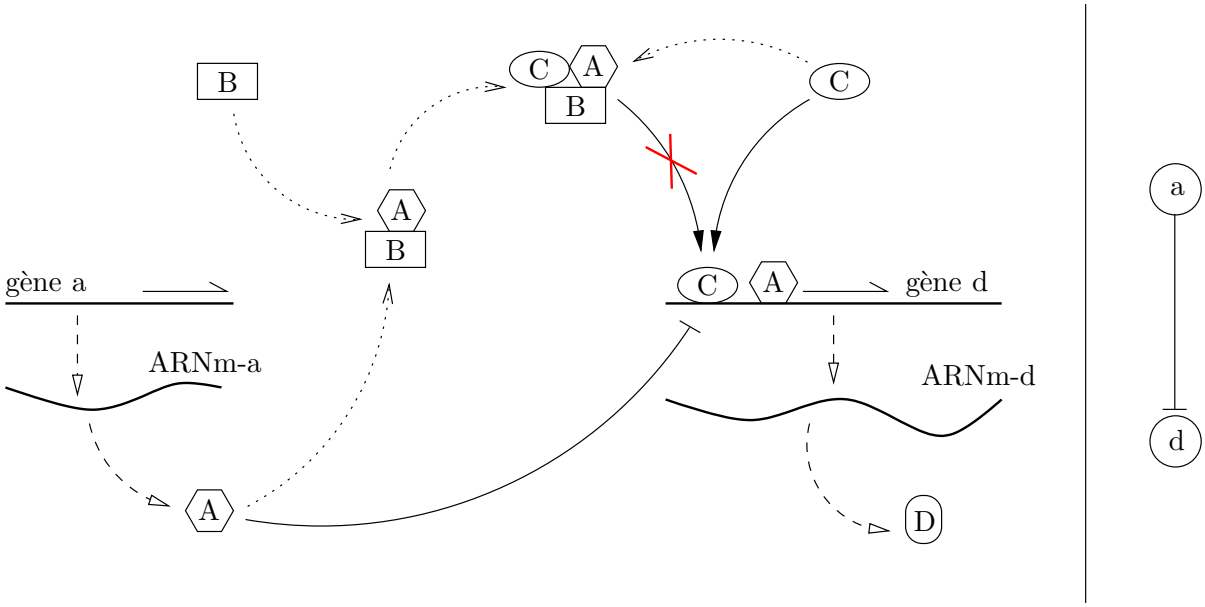


Fig. 1.5 – Différents mécanismes de régulation expliquant une régulation transcriptionnelle. À gauche un réseau de régulation biologique représente deux mécanismes de régulation pouvant expliquer l'inhibition du gène **d** par le gène **a** : le produit du gène **a** inhibe directement l'expression du gène **d** ; la protéine **C** qui active le gène **d** est capturée par un hétérodimère constitué du produit du gène **a** et d'une protéine **B**. À droite, le réseau de régulation transcriptionnelle qui modélise ces deux cas de figure.

l'intermédiaire d'une autre protéine.

Comme on peut le constater, les RRT comportent un grand nombre d'hypothèses simplificatrices qui confèrent à cette approche une part importante d'incertitude. Ils permettent cependant de mener une étude systémique de la cellule à travers l'un de ses principaux processus de contrôle : la transcription. Cette approche s'appuie sur une méthode expérimentale puissante (l'étude des profils d'expression des gènes), et des technologies associées (puces à ADN) qui se sont considérablement développées au cours des dernières années.

Chapitre 2

Les outils d'étude de la génomique fonctionnelle

Alors que le séquençage du génome touche à sa fin, la génomique fonctionnelle, c'est-à-dire l'étude et la caractérisation des fonctions des gènes dont les séquences sont désormais connues, constitue une nouvelle étape de la biologie moléculaire. En effet, un gène contient les instructions nécessaires à la synthèse d'un ARNm, cependant à un moment et dans une cellule donnée, seule une faible proportion des gènes sont utilisés pour synthétiser de l'ARNm. On dit alors de ces gènes qu'ils sont exprimés alors que les autres sont non exprimés. Par ailleurs les gènes exprimés le sont à des niveaux variables. Sous certaines conditions, ces niveaux d'expression peuvent être régulés. Quand c'est le cas, ils peuvent alors être induits ou réprimés, c'est-à-dire que leur niveau d'expression peut augmenter ou diminuer de manière plus ou moins importante. De nombreux facteurs externes (signaux environnementaux) ou internes (cellule en phase active de division ou dans une étape particulière du développement) décident du niveau d'expression des gènes exprimés. Les cellules musculaires, nerveuses, ou les cellules de la peau par exemple, diffèrent dans une large mesure par le sous-ensemble de gènes qu'elles expriment et par le niveau d'expression de ces derniers. Par conséquent, déterminer le mode d'expression des gènes doit permettre de caractériser le type des cellules ou la réponse de ces dernières à des signaux environnementaux. Pour obtenir cette information, on cherche généralement à mesurer des profils d'expression. Un profil d'expression rend compte du niveau de transcrit à l'équilibre pour des milliers de gènes dans une condition donnée, afin de produire une image globale de l'état de fonctionnement cellulaire.

La technologie la plus emblématique pour mesurer des profils d'expression est celle des puces à ADN permettant notamment de mesurer le transcriptome d'un échantillon de cellules. Typiquement, un stimulus permet de perturber un réseau de régulation génétique, par exemple en activant un gène qui normalement n'est pas exprimé, déclenchant ainsi une cascade de nouveaux effets d'expression. L'étude des profils d'expression obtenus pour une collection de perturbations judicieusement choisies doit ensuite permettre de capturer l'information relative au système régulatoire à l'œuvre dans la cellule.

Il existe une deuxième approche à grande échelle qui est l'étude du protéome, c'est-à-dire l'étude simultanée des concentrations des protéines d'un échantillon. Le génome humain contient de l'ordre de 25 000 gènes travaillant de concert pour produire de l'ordre de 1 000 000 protéines distinctes. Cela s'explique surtout par les importants changements opérés au sein des protéines

synthétisées lorsqu'elles subissent des modifications post-traductionnelles, de telle sorte qu'un gène sert de matrice pour de nombreuses versions distinctes d'une même protéine. Les méthodes permettant de mesurer les concentrations d'un ensemble de protéines au sein d'un échantillon donné, telle que la spectrométrie de masse, ne permettent de mesurer qu'une faible fraction des différentes protéines. En outre, ces données présentent encore d'importants problèmes d'exploitation et d'interprétation. Bien que l'étude du protéome semble plus pertinente que celle du transcriptome dans l'étude des réseaux de régulation biologique, les profils d'expression de gènes demeurent le moyen le plus efficace d'obtenir une image globale de l'activité cellulaire en un nombre restreint d'expériences.

2.1 Étude du transcriptome et technologie des puces à ADN

La possibilité de mesurer à l'échelle de la cellule l'abondance relative des transcrits a constitué une véritable révolution en génomique. Le développement des puces à ADN a permis d'augmenter fortement le débit des études d'expression des gènes. D'autres techniques, fondées sur des approches de séquençage des transcrits, existent. On peut citer la méthode de SAGE (*Serial Analysis of Gene Expression*) qui a l'avantage de ne pas se limiter à la détection des gènes dont la séquence est déposée sur la puce et qui par conséquent, est dite non biaisée. À l'heure actuelle, les puces à ADN restent la technologie la plus employée pour mener ce type d'étude, même si les nouvelles générations de séquenceurs rapides nous permettent de penser que les approches de type SAGE pourront dans un avenir proche être mis en œuvre beaucoup plus systématiquement.

2.2 Une introduction aux puces à ADN

2.2.1 Description et applications

Une puce à ADN est un support solide (classiquement une lame de verre, de plastique ou une puce en silicone) à la surface duquel de courtes séquences d'ADN communément appelées *sondes* (ou séquences reportrices) sont fixées via une liaison covalente à une matrice chimique. Les sondes de même séquence sont réparties au sein de spots (des portions de surface microscopiques) disposés de manière ordonnée et uniforme sur la puce. Chaque spot peut contenir des milliers d'exemplaires d'une unique séquence d'ADN qui lui est spécifique. Les sondes s'hybrident spécifiquement avec leur ARN (ou ADN) cible par appariement de séquences. Les molécules ciblées sont présentes au sein d'un échantillon d'ARN ou d'ADN à analyser. La plupart du temps, la détection des acides nucléiques hybridés est assurée à l'aide d'un marquage fluorescent préalable des ARN (ou ADN) de l'échantillon. Ainsi, les sondes permettent de mesurer la quantité relative d'une séquence d'ARN (ou d'ADN) entre deux échantillons en utilisant un marquage différentiel [1] ou des puces distinctes [2]. Dans la mesure où une puce à ADN peut contenir des dizaines de milliers de spots, son utilisation permet potentiellement d'accomplir un nombre équivalent de mesures en parallèle.
Les puces à ADN sont communément utilisées pour la détection d'ARN (et plus généralement d'ADN complémentaires obtenus par rétro-transcription) pour faire de l'analyse d'expression. Elles ont cependant permis d'accélérer significativement un grand nombre de recherches. On peut citer les applications suivantes.

[1] C'est le cas des puces à deux couleurs où les deux échantillons à comparer sont marqués avec deux fluorophores différents.

[2] C'est le cas des puces à oligonucléotides courts, telles que les puces Affymetrix.

Analyse d'expression de gènes Dans un profil d'expression de gène, les niveaux d'expression de milliers de gènes sont suivis simultanément pour étudier les effets de certains traitements, maladies, ou étapes de développement, sur l'expression des gènes. Par exemple, il s'agira d'identifier les gènes dont l'expression est modifiée en réponse à des pathogènes en comparant l'expression des gènes dans des tissus infectés et non infectés.

ChIP on Chip (*Chromatin ImmunoPrecipitation on Chip*) Les portions de séquence d'ADN sur lesquelles est fixé un facteur de transcription peuvent être isolées au moyen d'une technique appelée *immunoprécipitation* (ChIP). Brièvement, après pontage covalent des protéines liées à l'ADN, la chromatine est extraite puis fragmentée par sonication ou digestion enzymatique. On procède ensuite à la sélection des segments d'ADN qui sont liés au facteur de transcription choisi en utilisant un anticorps contre cette protéine particulière : c'est le principe de l'immunoprécipitation. Après avoir détaché les protéines, les morceaux de chromatine sont amplifiés et marqués avec un fluorochrome. Les séquences d'ADN marquées peuvent alors être hybridées sur une puce à ADN [RRW$^+$00] afin de déterminer les sites de fixation de la protéine d'intérêt à travers le génome.

Les puces de génotypage Les puces à ADN peuvent également être utilisées pour parcourir le génome dans sa globalité afin d'identifier des variations génétiques (de séquences) en différents points du génome.

Le ChIP-on-chip est une technique permettant d'interroger un facteur de transcription sur ses cibles génomiques à un moment donné. Cette information est d'une extrême importance, puisque contrairement aux approches in silico de recherche de sites de fixation, elle correspond à une fixation effective des facteurs de transcription sur leurs sites. Cette méthode apparaît donc comme complémentaire de l'analyse d'expression dans l'étude des systèmes régulatoires. Elle soulève cependant deux problèmes. Le premier concerne la localisation des sites de fixation des facteurs de transcription par rapport à leur gène cible. Les connaissances récemment acquises sur la régulation de la transcription chez les eucaryotes ont montré que les sites de régulation d'un gène sont éparpillés, et parfois même relativement éloignés du gène cible. Se pose donc la question de savoir à quelle gène cible correspond un site de régulation. Un autre problème concerne la distorsion entre la fixation d'un facteur de transcription sur un site de régulation et l'impact que cela peut avoir sur l'expression du gène cible. En effet, la fixation d'un facteur de transcription sur l'ADN peut ne pas influencer le niveau d'expression du gène : tous les sites de fixation ne sont pas des effecteurs, et certains ne le sont que dans certaines circonstances. Ces résultats d'expériences de ChIP-on-chip doivent ainsi être interprétés avec les précautions qui s'imposent.

2.2.2 Protocole générique d'une expérience de puce à ADN

Nous présentons dans ce qui suit une expérience générique de puce à ADN. On se place ici dans le contexte de l'analyse d'expression, bien que l'étape 2 mette en relief la façon dont on peut utiliser cette technique pour une étude des cibles des facteurs de transcription (ChIP-on-chip).

1. Acquisition des deux échantillons à comparer. Il s'agira typiquement d'un échantillon traité (test) et non traité (contrôle).
2. Les acides nucléiques d'intérêt sont purifiés. Pour une analyse d'expression de gène il s'agit de l'ensemble des ARNm, alors que pour une étude de régulation (ChIP-on-chip), il s'agit des ADN/ARN qui étaient liés à une protéine particulière et qui ont été sélectionnés par immunoprécipitation.
3. Les produits marqués sont générés par transcription inverse. Il s'agit du mécanisme moléculaire par lequel un brin d'ADN (dit ADN complémentaire et noté ADNc) est synthétisé à partir

d'un brin d'ARN. L'ADNc produit présente en effet une séquence complémentaire de celle du gène codant pour l'ARN rétro-transcrit. Sans aller dans les détails, l'étape de synthèse des ADNc permet également d'y incorporer l'un des deux fluorochromes évoqués plus tôt : Cyanine 3 (Cy3) ou Cyanine 5 (Cy5).

4. Dans le cas d'une puce à deux couleurs, le mélange des échantillons marqués est ensuite déposé sur la puce et mis à hybrider après dénaturation des acides nucléiques (les repliements 2D et 3D éventuels des acides nucléiques d'intérêt sont brisés, au même titre que les appariements de séquence). Pour les puces à une couleur, chaque échantillon est hybridé sur une puce différente.
5. Après une nuit d'hybridation, la puce est lavée afin d'éliminer tous les appariements non spécifiques ainsi que toutes les molécules non appariées.
6. Elle est ensuite séchée puis placée dans un scanner, où des lasers (préalablement réglés aux longueurs d'ondes d'excitation des fluorochromes employés pour le marquage) excitent les acides nucléiques marqués et hybridés. La fluorescence émise par ces molécules est ensuite enregistrée par des capteurs, pour chacune des deux longueurs d'onde d'émission des deux fluorochromes, générant ainsi deux images (puces deux couleurs). Si un seul fluorochrome est utilisé, une seule image est générée (puces Affymetrix) au niveau de chaque spot.

Un schéma résumant ce protocole est présenté à la figure 2.1.

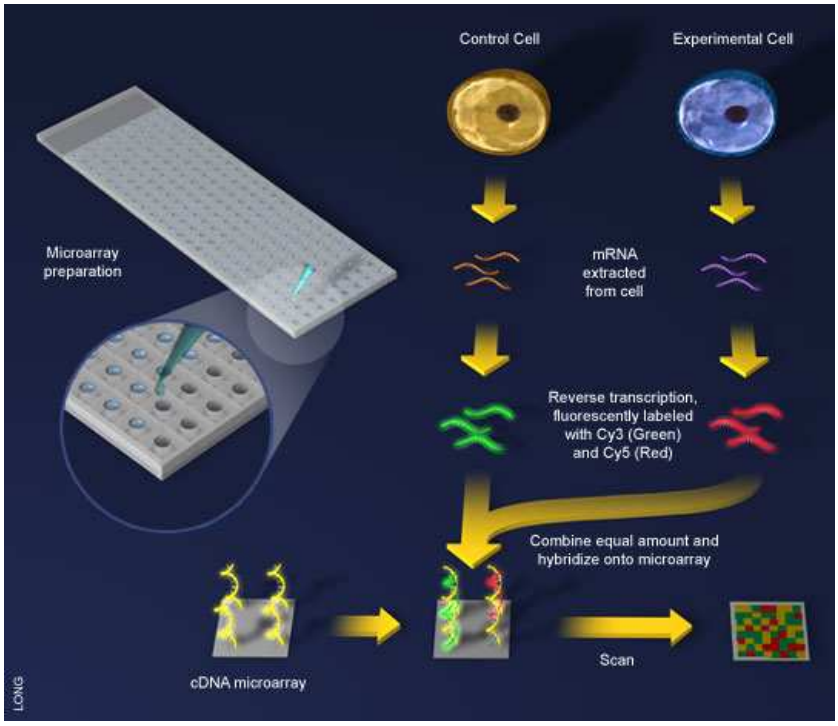


Fig. 2.1 – Schéma du principe des puces à ADN en double couleur.
Source : http ://www.scq.ubc.ca/

Bien que la description qui vient d'être faite soit *a priori* indépendante du type de puce utilisé, nous allons voir à présent qu'il existe différent types de puces permettant de réaliser ce type d'étude de différentes manières.

2.2.3 Les différents types de puces à ADN

Les puces à ADN peuvent être fabriquées de différentes manières. La méthode de conception retenue dépend essentiellement de considérations budgétaires, des besoins (parfois très spécifiques) des expérimentateurs, ainsi que de la nature de la question biologique qui est posée. Un rapide tour d'horizon des différents types de puces à ADN disponibles est utile afin de comprendre la grande hétérogénéité des outils mis en œuvre et incidemment, des données collectées. Prendre conscience de ce fait permet de mieux appréhender la principale difficulté de cette famille de techniques expérimentales : la difficulté à réunir un nombre important de mesures cohérentes afin de mener une étude statistique du transcriptome pour un phénomène donné. Dans un premier temps, nous distinguerons les puces selon que les sondes présentes à leur surface sont spottées (déposées) ou synthétisées au sein même de la puce (synthèse *in situ*). Par abus de langage, on parle fréquemment de *puces spottées* et de puces *in situ*. Nous détaillerons ensuite, le principe des puces en deux couleurs et mono-couleur.

2.2.3.1 Les puces spottées

Les sondes présentes sur les puces spottées sont le plus souvent des oligonucléotides ou des ADNc. Elles sont synthétisées préalablement à leur dépôt sur la puce et sont ensuite *spottées* (déposées) à la surface de la lame. La méthode classique consiste à utiliser des aiguilles très fines, contrôlées par un robot qui dépose chaque type de sonde à intervalle régulier à la surface de la puce. Il en résulte une grille dont chaque élément, repéré par ses coordonnées (numéro de ligne et de colonne), est prêt à s'hybrider à la séquence complémentaire des ADNc ou ARN cibles extraits d'échantillons expérimentaux. Cette technique permet de concevoir des puces « à façon », dans la mesure où les chercheurs peuvent aisément adapter le contenu des puces produites (nature et position des sondes) aux besoins spécifiques de leurs expériences. Ils peuvent également contrôler toute la chaîne de traitement dans la mesure où ils peuvent produire les puces et les échantillons marqués à hybrider, réaliser l'hybridation et enfin scanner les puces obtenues avec leur propre matériel. En règle général, cette approche permet de minimiser les coûts d'une expérience de puce à ADN de manière significative. En effet, les puces du commerce sont souvent nettement plus chères et contiennent généralement un nombre très important de sondes dont la majeure partie n'intéresse pas nécessairement l'expérimentateur. Bien que cette question soit sujette à controverse, il semble cependant que les puces commerciales produites selon des processus industriels optimisés offrent une sensibilité supérieure aux puces maison. Toutefois, il va de soi que la qualité des puces spottées dépend avant tout du laboratoire producteur, du savoir-faire de ses personnels et des performances du matériel utilisé. Plus spécifiquement, elle est fortement déterminée par l'efficacité des procédures de production mises en place en laboratoire et les protocoles expérimentaux de préparation des ARN (ou ADNc) et d'hybridation.

2.2.3.2 Les puces *in situ*

Le terme de « puces *in situ* » fait référence à la technique de production utilisée, les sondes étant directement synthétisées sur la surface de la puce par un procédé physico-chimique appelé *photolithographie*. De ce fait, la taille des sondes produites est plus faible et leur densité sur la puce est plus importante que pour des puces spottées. Plus précisément, ces sondes sont de courtes séquences d'ADN conçues pour s'hybrider avec des parties d'une séquence codante (ou d'une phase ouverte de lecture) et pour représenter un unique gène ou une famille de variantes d'épissage. La taille de ces séquences peut varier selon le but recherché : si des séquences plus courtes permettent d'avoir une densité de sonde plus importante sur la puce et sont moins chères à produire, des séquences plus longues sont plus spécifiques d'un gène cible. Typiquement,

les sondes synthétisées par Agilent sont des 60-mer alors que celles d'Affymetrix sont des 25-mer.

Nous faisons ensuite la distinction entre les puces à double couleur et les puces mono-couleur. Les différences soulignées ici concernent plus particulièrement le type de mesures réalisées sur un échantillon afin de bâtir un profil d'expression. On distingue grossièrement l'hybridation compétitive des puces à deux couleurs, qui permet de mesurer un différentiel d'expression entre deux conditions expérimentales sur une même puce, et l'hybridation simple des puces mono-couleur, qui est utilisée pour effectuer des comparaisons entre des puces différentes rendant chacune compte de l'expression des gènes dans une condition expérimentale particulière.

2.2.3.3 Puces à deux couleurs et puces mono-couleur

Typiquement, les puces à deux couleurs (ou puces à doubles canaux) sont hybridées avec des ADNc préparés à partir de deux échantillons distincts que l'on souhaite comparer (tissu malade versus tissu sein) et qui sont marqués avec deux fluorochromes différents : la Cyanine 3 (Cy3), qui a une longueur d'onde d'émission de 570 nm correspondant au vert dans le spectre lumineux, et la Cyanine 5 (Cy5), qui a une longueur d'onde d'émission de 670 nm et dont la fluorescence est rouge. Les deux échantillons d'ADNc ainsi marqués sont mélangés et hybridés sur la puce qui est ensuite scannée afin de visualiser la fluorescence des deux fluorochromes après excitation avec un rayon laser calibré aux longueurs d'ondes adéquates (570 puis 670 nm). Les intensités de fluorescence des deux fluorochromes peuvent ensuite être utilisées pour mener une analyse différentielle afin d'identifier les gènes induits ou réprimés entre les deux échantillons. On parle d'hybridation compétitive car les ADNc extraits des deux échantillons ayant (théoriquement) la même affinité pour une sonde donnée, leur probabilité d'hybrider cette sonde ne dépend que de leurs concentrations relatives dans les échantillons testés. Si un ARNm est deux fois plus concentré dans les tissus malades marqués en vert (avec la Cyanine 3) que dans les tissus sains marqués en rouge (avec la Cyanine 5), une sonde a une probabilité deux fois plus importante de fixer les ADNc marqués en vert que ceux marqués en rouge.

Les puces mono-couleur (ou puces mono-canal) sont conçues pour fournir une mesure du niveau d'expression des gènes dans un échantillon spécifique. Pour être interprétée, cette mesure doit être comparée à celle obtenue pour un échantillon de référence. Dans ce cadre, la comparaison de deux conditions expérimentales nécessite donc d'hybrider les deux échantillons correspondants sur deux puces distinctes avec un marquage unique à chaque fois. Le principal avantage de cette approche est de faciliter la comparaison des données de puces appartenant à des expériences distinctes. En effet, contrairement aux puces à deux couleurs, l'approche mono-couleur ne nécessite pas de procéder au mélange des échantillons à comparer qui fait que les deux mesures sont appariées inexorablement. Les mesures obtenues dans différentes conditions expérimentales peuvent alors être comparées deux à deux *a posteriori*, en fonction des besoins de l'expérimentateur : les résultats d'une puce **A** peuvent être comparés avec ceux d'une puce **B** un jour puis avec ceux d'une puce **C** obtenus le lendemain. Éventuellement, pour faciliter une comparaison globale entre différentes mesures on pourra, dans un premier temps, toutes les comparer à une référence commune. Si pour une unique comparaison le nombre de puces utilisées est donc deux fois plus important que pour des puces à double couleur, ce désavantage s'évanouit dès que le nombre de comparaisons devient plus élevé.

Les puces mono-couleur commerciales, telles que celles proposées par Affynetrix, présentent également un certain avantage en termes de qualité et de reproductibilité des résultats. Cela s'explique avant tout par le fait que les protocoles de production et d'hybridation sont fortement normalisés par rapport aux puces à deux couleurs. Le coût de ces puces commerciales, bien qu'il

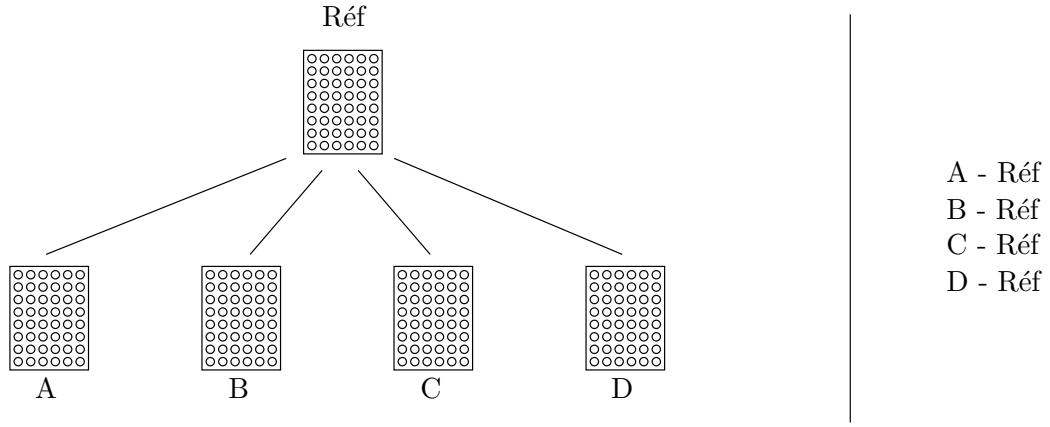


Fig. 2.2 – Schéma du dessin expérimental comparant des échantillons à une référence commune. À gauche, un graphe représente un ensemble d'échantillons d'ADN ou d'ARN obtenus dans différentes conditions expérimentales. Une arête entre deux échantillons signifie qu'ils sont hybridés sur une puce en double couleur. Dans ce cas, tous les échantillons sont comparés à un échantillon de référence. À droite, toutes les comparaisons directes représentées dans le graphe sont énumérées.

ait fortement été revu à la baisse récemment, est malheureusement souvent rédhibitoire pour de nombreux laboratoires. C'est pourquoi les puces spottées sont fréquemment utilisées pour faire de la comparaison de profils d'expression.

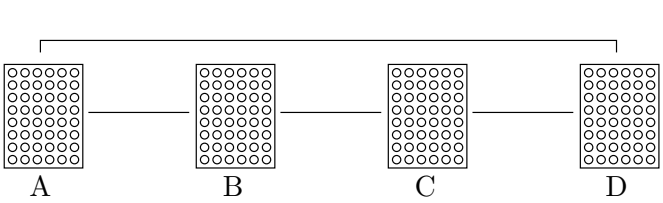
2.2.4 La conception des expériences de puces à ADN

Au-delà de la technologie employée, la manière même de concevoir une expérience de puces à ADN affecte à la fois l'efficacité ainsi que la validité de l'expérimentation. Plusieurs points peuvent être évoqués à ce sujet pour mettre en avant l'extrême hétérogénéité des expériences de puces à ADN et éclairer la grande variabilité des données collectées par cette technique.

Nous nous plaçons dans le cas d'une analyse d'expression génétique réalisée au moyen d'une puce en double couleur. Nous pourrions cependant aussi bien envisager l'utilisation d'un couple de puces mono-couleur pour comparer deux conditions expérimentales données. La première étape avant de concevoir une expérience de puce à ADN est de définir quelles sont les variétés d'ARN à comparer. Ce choix intervient dans l'élaboration du dessin expérimental. À ce titre, il existe principalement deux grandes familles de dessins expérimentaux : le dessin de référence et le dessin en boucle. Au-delà des différents types de comparaisons envisagées, un dessin expérimental peut également intégrer des aspects visant à faciliter l'estimation et la correction de certains biais tels que ceux introduits par les fluorochromes. Il permet également de mieux gérer la variabilité des données, que ses causes soient d'origine technique ou biologique.

2.2.4.1 Les principaux types de comparaisons d'échantillons

Dessin de référence Très utilisé dans la pratique, l'idée du dessin de référence représenté à la figure 2.2 est de comparer chaque variété d'ARN à une variété de référence. Tout couple d'échantillons peut donc être comparé par l'intermédiaire de la variété de référence, ce qui permet à toutes les comparaisons de présenter la même efficacité. Par ailleurs ce dessin garantit l'évolutivité de l'expérience car il est aisé d'y introduire de nouveaux échantillons que l'on sou-



A - B
B - C
C - D
D - A

Fig. 2.3 – Schéma du dessin expérimental en boucle. À gauche, un graphe représente un ensemble d'échantillons d'ADN ou d'ARN obtenus dans différentes conditions expérimentales. Une arête entre deux échantillons signifie qu'ils sont hybridés sur une puce en double couleur. Dans ce cas, les échantillons sont ordonnés et chaque échantillon n'est comparé qu'avec ses voisins. À droite, toutes les comparaisons directes représentées dans le graphe sont énumérées.

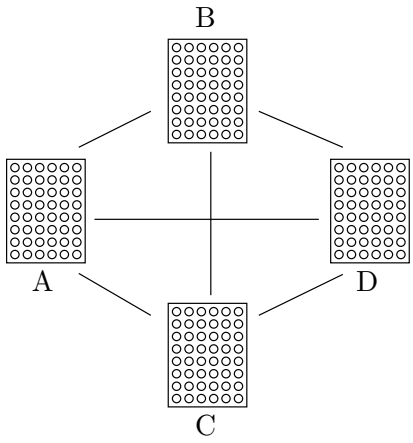
haite rajouter à l'étude. Il suffit pour cela de disposer de suffisamment d'ARN dans la variété de référence. C'est le type de dessin expérimental généralement utilisé lorsque l'on étudie des cinétiques d'expression : on mesure les profils d'expression d'échantillons de cellules prélevées à différents temps après avoir appliqué un stress (irradiation, choc thermique, utilisation de drogue, etc.) à l'organisme dont elles sont issues. Chaque mesure est alors comparée à celle effectuée au temps 0, c'est à dire lors de l'application du stress, ou un peu avant. C'est également le type de dessin expérimental utilisé dans le cadre d'une analyse de l'effet dose d'un stimulus. Par exemple, les profils d'expression de cellules soumises à des doses croissantes de radiation ou de drogue sont comparés au profil d'expression réalisé sur des cellules n'ayant pas subi le stimulus.

Dessin en boucle Le dessin en boucle représenté à la figure 2.3, consiste à comparer chaque variété d'ARN avec exactement une variété d'ARN distincte. Dans le cas de l'étude d'un effet dose comprenant D doses d'un drogue $\{d_1, d_2, \ldots, d_D\}$ correspondant chacune à un échantillon distinct, on a donc D paires du type $(d_i, d_{i+1}), \forall i \in \{1, 2, \ldots, D-1\}$ et (d_1, d_D). Ce type de dessin contient le même nombre de lames que le dessin de référence mais il collecte deux fois plus d'informations sur les variétés d'intérêt. On remarque également que la précision des comparaisons diminue avec la distance séparant deux échantillons le long de la boucle. Ce type de dessin n'est donc pertinent que pour un faible nombre d'échantillons.

Dessin de comparaison deux à deux. Il existe d'autres types de dessins expérimentaux. On peut par exemple envisager de comparer tous les échantillons deux à deux comme cela est représenté dans la figure 2.4. On augmente alors de manière significative le nombre de puces utilisées ($n(n-1)/2$ contre n auparavant, avec n le nombre de variétés d'ARN étudiées) mais on améliore sensiblement la qualité des comparaisons. Cette stratégie alternative ne réalisant que des comparaisons directes entre variétés d'ARN, les différences d'expression entre les variétés comparées peuvent être estimées avec une plus grande précision.

2.2.4.2 Les méthodes pour étudier la variabilité des données

Afin de limiter les problèmes relatifs à la variabilité des données, différentes approches expérimentales ont été proposées.



A - B
A - C
A - D
B - C
B - D
C - D

Fig. 2.4 – Schéma du dessin expérimental comparant des échantillons deux à deux. À gauche, un graphe représente un ensemble d'échantillons d'ADN ou d'ARN obtenus dans différentes conditions expérimentales. Une arête entre deux échantillons signifie qu'ils sont hybridés sur une puce en double couleur. Dans ce cas, tous les échantillons sont comparés deux à deux. À droite, toutes les comparaisons directes représentées dans le graphe sont énumérées.

La réplication d'expériences La *réplication technique* permet d'estimer et de réduire les effets de la variabilité imputable à la mesure de la fluorescence et en amont, à l'exécution du protocole expérimental. Elle consiste à hybrider les ARN issus d'un unique échantillon biologique sur des lames distinctes. Ces ARN peuvent avoir été obtenus par des extractions indépendantes ou être issus d'aliquots distincts d'une même extraction. Le *Dye-Swap* par exemple, vise à éviter tout biais dans la mesure des intensités de fluorescence dû aux fluorochromes. Pour cela, le dessin expérimental doit être équilibré vis-à-vis de ces derniers, c'est-à-dire que chaque échantillon d'ARN doit être étiqueté autant de fois avec le fluorochrome rouge qu'avec le fluorochrome vert. Chaque comparaison est donc généralement réalisée en deux exemplaires, les marquages employés étant intervertis d'une comparaison sur l'autre.

La *réplication biologique* consiste à hybrider des ARN issus de différents échantillons biologiques soumis au même stimulus (par exemple, plusieurs individus traités avec la même drogue) sur plusieurs puces. Les réplicats biologiques permettent d'étudier la variabilité biologique (entre échantillons), ainsi que la variabilité inhérente à la mesure.

Le regroupement d'échantillons biologiques La variabilité au sein des puces à ADN peut également être réduite en regroupant les ARNm obtenus à partir des réplicats biologiques au sein de pools. Par exemple, 20 cas divisés en 5 pools de 4 (chaque pool étant hybridé sur une puce différente) devraient avoir plus de puissance que 5 cas hybridés sur des puces différentes. Bien sûr, la puissance* demeurera inférieure à celle obtenue avec 20 cas hybridés séparément. Cette stratégie présente l'avantage d'accroître la taille de l'échantillon sans qu'il soit nécessaire d'utiliser un plus grand nombre de puces. Elle n'est cependant pas dépourvue d'inconvénients :

- dans l'éventualité d'un cas corrompu, on empoisonne le pool entier qui le contient ;
- les mesures effectuées sur un pool ne correspondent pas nécessairement à la moyenne mathématique des mesures des individus composant ce pool.

Cette approche expérimentale peut malgré tout s'avérer intéressante lorsque l'on souhaite effectuer une analyse différentielle sur des échantillons présentant une variabilité biologique élevée

en comparaison de l'erreur de mesure, et que ces échantillons biologiques sont peu coûteux au regard du prix des puces.

Limiter l'influence des facteurs extérieurs D'une manière générale, les mesures de puces à ADN peuvent être fortement influencées par des facteurs extérieurs. Si ces derniers covarient avec les variables indépendantes de notre étude (par exemple, avec les traitements appliqués à différents groupes de patients) cela peut conduire à des résultats erronés. C'est pourquoi il est crucial que de tels facteurs soient minimisés voire (idéalement) éliminés. Pour cela, les puces peuvent par exemple être hybridées et analysées par un seul technicien le même jour. Bien qu'il soit difficile de procéder de la sorte pour des expériences comportant un grand nombre de puces, il est malgré tout possible d'orthogonaliser ces facteurs. On pourra, par exemple, analyser un nombre identique d'échantillons pour chacun des groupes en cours d'évaluation chaque jour d'analyse.

2.3 Analyse des données de puces à ADN

Dans la section précédente, nous avons présenté le principe ainsi que les approches à la fois techniques et expérimentales qui sous-tendent les études de profils d'expression par puce à ADN. À présent, nous envisageons les méthodes de traitement permettant de tenir compte de la variabilité et de la complexité intrinsèque (du fait notamment du nombre de mesures simultanées mises en jeu) des profils d'expression en vue de leur interprétation [ACPS06]. Dans un premier temps, nous allons rapidement aborder la question du traitement des données brutes des puces à ADN, c'est-à-dire la façon de rendre des mesures de fluorescence exploitables. Nous évoquerons ensuite quelques unes des méthodes statistiques ou informatiques permettant d'en extraire de l'information.

2.3.1 Pré-traitement des données brutes de puces à ADN

Le pré-traitement des données brutes, qui inclut l'analyse d'image, la normalisation et la transformation [3] des données, demeure un champ de recherche relativement actif. Nous allons présenter ces différentes étapes sans nous attarder sur les aspects techniques. Notre but ici est de mettre en lumière la nature des informations obtenues après traitement des données brutes, qui sont généralement utilisées pour extraire de la connaissance.

2.3.1.1 Le traitement d'image

Dans un premier temps, on souhaite quantifier de manière appropriée la fluorescence de chaque spot sur la puce (voir les figures 2.6 et 2.5). Concrètement, il s'agit de convertir une image de puce représentant les intensités de fluorescence émises dans un canal d'émission au niveau de chaque spot en un tableau de données assignant à chaque gène une mesure rendant compte de son niveau d'expression. Il s'agit également de corriger le bruit de fond pouvant perturber ces mesures.
La principale différence entre les nombreuses approches ayant été développées dans ce but [YBDS00, SWS+01, EBKR04] concerne la façon de réaliser la segmentation des spots [4]. D'une manière générale, on peut distinguer deux étapes durant le traitement d'image.

[3] Application d'une fonction mathématique spécifique afin de changer la forme des données. Souvent, la nouvelle forme des données satisfait les hypothèses nécessaires à la mise en œuvre d'un test statistique. La transformation la plus courante dans les puces à ADN est le log_2.

[4] La segmentation consiste à séparer spatialement les différents spots du fond de l'image.

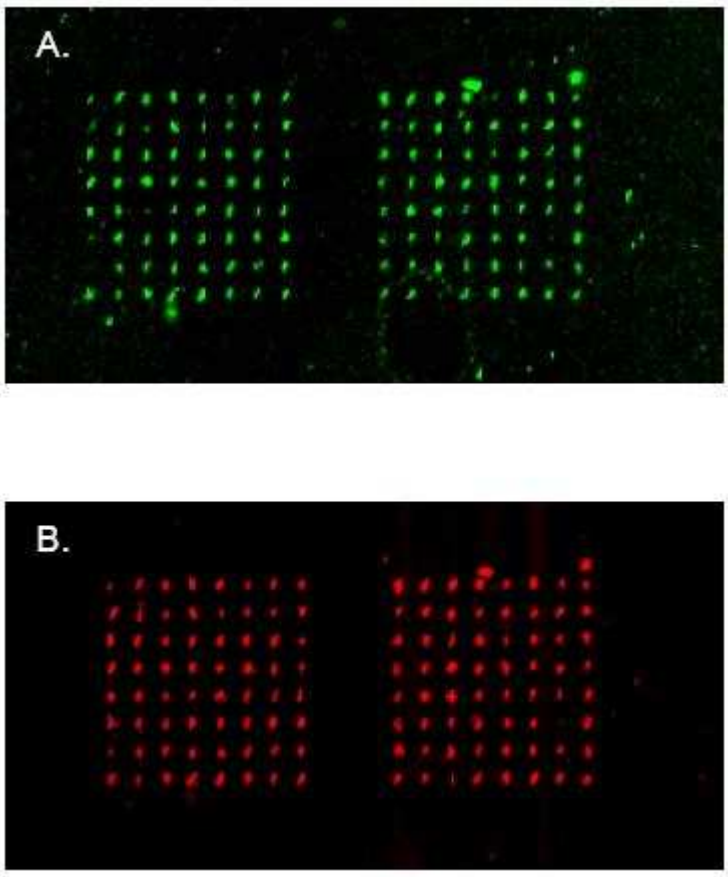


Fig. 2.5 – Images d'une puce à ADN en double couleur. Les intensités de fluorescence émises dans les deux canaux sont présentées séparément. Pour chaque spot nous avons : A- Intensité de fluorescence des acides nucléiques marqués avec la Cyanine 3 (Cy3) ; B - Intensité de fluorescence des acides nucléiques marqués avec la Cyanine 5 (Cy5).

1. **Segmentation des spots** Chaque spot étant le résultat de manipulations expérimentales, sa forme ainsi que sa superficie exacte peut varier d'une expérience à l'autre. Il s'agit donc de déterminer quels pixels font partie du spot et quels sont ceux qui appartiennent à son environnement immédiat. Il est ainsi possible de séparer l'image du spot de l'image de fond représentant le support de la puce. Cette segmentation peut être guidée par l'application d'une grille préalable traduisant la connaissance *a priori* que l'on a de la scène à analyser. Compte tenu du plan de dépôt on peut en effet rechercher les $M \times N$ blocs contenant chacun l'un des $m \times n$ spots. Les variantes de segmentation sont les suivantes.
    1. Recherche de formes prédéfinies pour des spots avec ou sans optimisation des paramètres de ces formes (comme le diamètre d'un disque par exemple).
    2. Recherche d'un spot de forme quelconque.
2. **Extraction des intensités** On peut ensuite mesurer l'intensité du spot que l'on a identifié et isolé. Pour cela on peut prendre la moyenne ou la médiane (cette dernière est généralement préférée) des pixels constituant un spot. Le but étant de mesurer l'intensité effectivement due à l'hybridation compétitive des brins d'ADNc marqués, on peut sous-

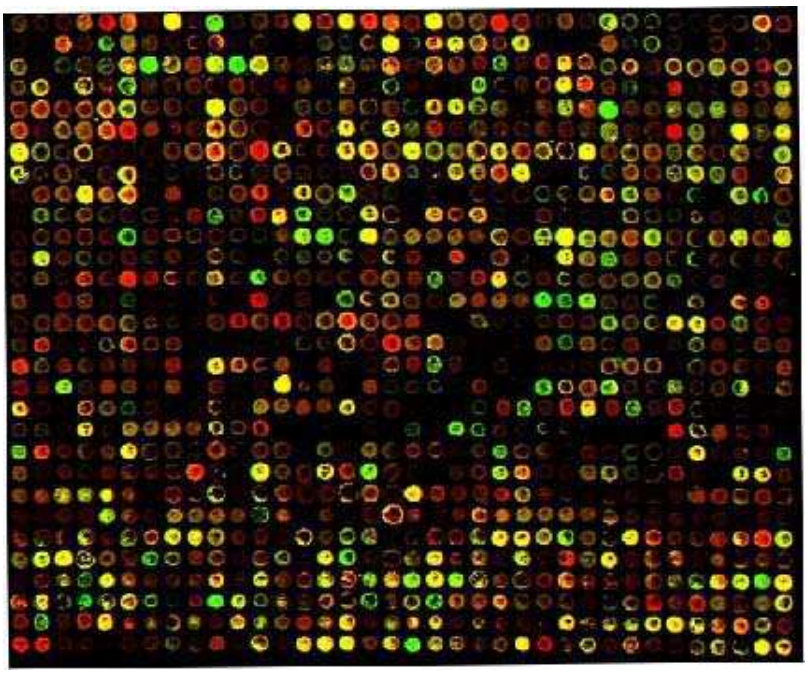


Fig. 2.6 – Une image de puce à ADN en double couleur après superposition des intensités de fluorescence des deux fluorochromes. On compare un échantillon A dont les acides nucléiques sont marqués avec la Cyanine 3 (Cy3), un fluorochrome « vert », et un échantillon B dont les acides nucléiques sont marqués avec la Cyanine 5 (Cy5), un fluorochrome « rouge ». Les intensités de fluorescence émises dans les deux canaux d'émission sont superposées. Seule une fraction des spots de la puce est représentée. Qualitativement, les spots de couleur verte correspondent aux acides nucléiques dont la concentration est plus élevée dans l'échantillon A que dans l'échantillon B. Inversément, les spots de couleur rouge correspondent aux acides nucléiques dont la concentration est plus élevée dans l'échantillon B que dans l'échantillon A. Les spots correspondant aux acides nucléiques dont la concentration est proche dans les deux échantillons apparaissent en jaune. Les spots sombres, présentant un signal plus faible, peuvent s'expliquer par une faible quantité d'acides nucléiques marqués dans les deux échantillons.

traire le bruit de fond (le support de la puce présentant parfois une fluorescence résiduelle) et estimer une seuil admissible de rapport signal/bruit pour considérer ou non les mesures.

2.3.1.2 La normalisation

L'autre grande étape de pré-traitement des données est la normalisation. Il s'agit du procédé par lequel les intensités des spots de puces à ADN sont ajustées pour prendre en considération la variabilité au sein de différentes expériences et plates-formes [5]. La normalisation permet notamment de corriger le biais lié à la mesure de l'intensité de fluorescence. Ce biais de mesure est lui-même provoqué par divers phénomènes qu'il n'est pas toujours possible de distinguer :

– la variabilité en abondance de matériel génétique entre plusieurs échantillons ;

[5] Une plate-forme est un type de puces à ADN défini par la conception de la puce (nature et répartition des sondes à sa surface) ainsi que la technique de fabrication utilisée.

- la variabilité en termes d'efficacité de marquage ou d'hybridation. Les deux fluorochromes peuvent s'incorporer plus ou moins bien au sein des acides nucléiques que l'on souhaite marquer. En outre, ils peuvent modifier différemment l'efficacité d'hybridation des ADNc auxquels ils sont liés ;
- la sensibilité des capteurs qui dépend grandement de la façon dont le scanner est calibré. Par ailleurs, dans les cas limites des très grandes ou très faibles intensités, on constate fréquemment une augmentation de l'erreur sur la mesure.

Pour chacun des ADNc étudiés, on espère ainsi obtenir une mesure d'intensité de fluorescence qui rende compte du niveau d'expression du gène correspondant, indépendamment des autres effets susceptibles d'être mesurés. La normalisation permet de comparer des expériences de puces à ADN distinctes et de contrôler les variations indésirables entre les expériences. En général, elle rend également les données plus cohérentes avec les hypothèses sous-jacentes à de nombreuses procédures d'inférence.
Une approche simple pour décider de la méthode de normalisation à employer consiste à tracer un MA-plot. Il s'agit d'une figure représentant chaque gène selon les coordonnées $M = \log_2(R/V)$ et $A = \frac{1}{2}\log_2(R \times V)$, R et V étant respectivement l'intensité du marqueur rouge et vert pour le gène en question. Si on fait l'hypothèse (communément admise) que la majorité des gènes sont invariants dans une expérience donnée, alors la majorité des points devraient se trouver à proximité de l'axe $M = 0$. L'allure du MA-plot et son décalage avec la représentation graphique attendue sous l'hypothèse précédemment citée permet de choisir une correction adéquate. De nombreuses approches de normalisation ont été introduites et sont discutées dans [Qua02, YDL+02, SYS03]. La plupart des algorithmes de pré-traitement ont été développés et évalués en utilisant un ou deux jeux de données de petite taille avec un seul type de puces, chez l'humain le plus souvent. On peut donc craindre que de telles méthodes soient optimisées pour ces jeux de données et donnent de moins bons résultats dans d'autres conditions. D'une manière générale, les algorithmes de traitement d'image et de normalisation abondent et varient substantiellement dans leur approche.

Afin d'insister sur la multiplicité des méthodes visant à assurer la robustesse des données, nous allons finir ce descriptif par un cas particulier concernant les puces à oligonucléotides fabriquées par la société Affymetrix. Ces puces mono-couleur présentent une originalité en termes de conception : pour chaque gène cible, deux jeux de sondes de 25 nucléotides chacune ont été déposés sur la puce :

- des sondes notées PM (pour *Perfect Match*) spécifiques d'un transcrit ;
- des sondes notées MM (pour *MisMatch*) identiques aux premières sauf au niveau de la 13^{e} paire de base qui a été volontairement mutée.

Une sonde PM s'hybridant spécifiquement avec un transcrit donné peut également s'hybrider avec des transcrits non spécifiques mais présentant malgré tout des homologies de séquence avec cette sonde. Ce phénomène, appelé *hybridation non spécifique*, est mesuré à l'aide des sondes MM. Le pré-traitement de ces puces à oligonucléotides comprend donc une étape de correction du phénomène d'hybridation non spécifique permettant essentiellement de compenser le biais dû à ce phénomène [IWJ06].

2.3.2 Analyse différentielle

L'analyse des données d'expression se ramène presque toujours à une comparaison entre des niveaux d'expression mesurés dans deux conditions expérimentales distinctes. Compte tenu de la variabilité de ces mesures, on recourt classiquement à l'inférence statistique. Celle-ci consiste à tirer des conclusions quant à la véracité d'hypothèses concernant des caractéristiques non

observées d'une population globale. Elle est fondée sur des statistiques issues d'échantillons de cette population. Dans notre cas, ces échantillons (biologiques ou cliniques) sont représentés par les puces à ADN à notre disposition. On pose une hypothèse nulle : « la moyenne de l'expression d'un gène donné pour les individus soumis à un traitement A est différente de celle des individus soumis à un traitement B dans la population théorique de tous les patients susceptibles d'avoir été traités ». Plus généralement, il s'agit d'identifier les gènes différentiellement exprimés entre deux situations expérimentales au moyen de méthodes permettant de minimiser les erreurs de type I* et de type II* . Nous ne discutons pas ici de tests spécifiques en détail. Nous discutons de points qui sont communs à la plupart des méthodes d'inférence statistique utilisant par exemple la statistique t (Student), la statistique de Fisher ou la régression logistique.

2.3.2.1 Analyse différentielle et tests statistiques

La plupart du temps, on est amené à calculer les ratios d'expression de chaque gène entre deux conditions expérimentales d'intérêt (test et contrôle). On utilise couramment le terme *Fold Change*, noté FC, pour désigner la valeur de ce ratio qui constitue une mesure raisonnable de la taille de l'effet étudié (du changement d'expression). Le FC fut la première méthode utilisée pour évaluer si un gène est différentiellement exprimé. La popularité du FC provient tout d'abord de sa simplicité. On a fréquemment considéré qu'un gène est régulé (car différentiellement exprimé) à partir du moment où le ratio de ses niveaux d'expression dépasse 2. Cette méthode apparaît aujourd'hui discutable sur le plan statistique dans la mesure où elle ne tient aucun compte de la variabilité des données, de la taille de l'échantillon et n'associe aucun niveau de confiance aux conclusions tirées [BSSS03]. Par exemple, pour un petit nombre de réplicats (typiquement de l'ordre de 5), il suffit d'un seul cas présentant une forte disparité avec les autres mesures pour tirer la valeur moyenne du ratio observé au-delà du seuil préalablement fixé. Du reste, le fait que la mesure de l'expression d'un gène varie d'une condition à l'autre (test et contrôle) peut certes s'expliquer par le facteur étudié, mais aussi par d'autres facteurs extérieurs tels que l'état des cellules lors de l'extraction des ARN ou bien la façon dont l'expérience a été menée. Surtout, le simple fait de fixer arbitrairement un seuil au-delà duquel on estime qu'un ratio d'expression traduit un phénomène de régulation n'a pas véritablement de sens sur le plan biologique et ne constitue pas une approche statistiquement fondée.
Pour s'affranchir de ces problèmes, il est courant de recourir à des statistiques prenant en considération à la fois les ratios d'expression et la variabilité des données afin d'attribuer une valeur de significativité ou un intervalle de confiance à un résultat. On peut citer la statistique t (Student), la statistique z ou la statistique de Fisher (souvent dénommée *analyse de la variance* ou ANOVA) par exemple. La plupart du temps, ces statistiques prennent en compte les variations entre les réplicats correspondant aux variations que nous ne souhaitons pas mesurer, car indépendantes des facteurs étudiés. La mesure de significativité (*p-valeur*) permet d'estimer la probabilité pour que les effets observés dans les données (par exemple, une expression différentielle) soient le fruit du hasard. L'expérimentateur fixe donc une borne supérieur à la p-valeur afin de contrôler le nombre d'erreurs de type I. Plus cette borne est faible (ses valeurs variant généralement entre $0,05$ et $0,001$), plus grande est la confiance dans le résultat. Les techniques citées ici font références à des statistiques paramétriques. Elles reposent sur des conjectures quant à la distribution des variables étudiées et dérivent des propriétés des distributions théoriques pour faire de l'inférence. Il est aussi classique de mettre en œuvre des statistiques non paramétriques comme les statistiques de rang [TTC01]. Ces approches non paramétriques ne reposent pas sur un modèle statistique et présentent l'avantage d'être suffisamment robustes et flexibles pour traiter une nouvelle statistique sans qu'il soit nécessaire de dériver mathématiquement la distribution de cette dernière.

Le ré-échantillonnage des données est une alternative pour estimer la significativité des tests statistiques. Il sert à la fois pour les approches de tests paramétriques et non paramétriques. Une méthode classiquement utilisée est le *bootstrap*. Il consiste à construire des pseudo-jeux de données par échantillonnage des données réelles avec remplacement. On peut ainsi étudier la variabilité d'une méthode statistique appliquée à ces différents jeux de données artificiels. L'inconvénient majeur de cette approche est son coût de calcul important.

2.3.2.2 Le problème des comparaisons multiples

Compte tenu du grand nombre de gènes présents sur une simple puce, la mise en œuvre d'approches de tests d'hypothèses nécessite de prendre en compte un problème de comparaisons multiples. Le test multiple, c'est-à-dire le fait de tester de multiples hypothèses (une par gène sur la puce) au sein d'une même étude, présente des problèmes important : même si la p-valeur affectée à un gène donné indique qu'il est très peu probable que le différentiel d'expression de ce gène soit dû au hasard plutôt qu'à l'effet du traitement étudié, le nombre très élevé de gènes sur la puce rend vraisemblable que l'expression différentielle de certains gènes soit des faux positifs. Prenons un exemple simple.

Exemple 2.1
Une p-valeur de 0,05 est interprétée comme une mesure de significativité qui estime à 5% la probabilité d'observer le différentiel d'expression d'un gène par hasard. Avec une puce permettant de mesurer l'expression de 10 000 gènes et en utilisant une mesure de significativité $p < 5\%$, nous pouvons identifier jusqu'à 500 gènes significativement modulés, même en l'absence de toute dérégulation effective.

Pour limiter cet effet indésirable il est possible d'utiliser un critère de sélection plus sévère, en l'occurrence une p-valeur plus faible. Il existe également des méthodes plus élaborées telles que la correction de Bonferroni qui adapte le niveau de la borne supérieure imposée à la p-valeur pour chaque test. Elle vise à contrôler un indice appelé *family-wise error rate* ou FWER correspondant à la probabilité de faire au moins une erreur de type I parmi les N tests effectués (N étant le nombre total de gènes présents sur la puce). L'inconvénient majeur de cette technique est qu'elle est trop conservatrice. En restreignant le taux de faux positifs, elle augmente considérablement le taux de faux négatifs : de nombreux gènes différentiellement exprimés ne sont pas reconnus comme tels. Les biologistes sont pourtant prêts à tolérer l'existence de telles erreurs dans la mesure où cela permet de réaliser des découvertes. Par exemple, un chercheur peut juger acceptable qu'une proportion (raisonnable) de ses découvertes soit erronées (de l'ordre de 10%). Cette différence entre les attentes des biologistes et les outils fournis par les méthodologistes a donné naissance à de nouvelles approches pour l'inférence. Benjamini et Hochberg [BH95] ont défini le *taux de faux positifs* (*False Discovery Rate* [6]), noté FDR, et proposé des procédures permettant de le contrôler. Il s'agit d'une méthode de comparaison moins conservatrice, garantissant une plus grande puissance statistique que le contrôle du FWER décrit précédemment.
Les statistiques actuellement utilisées visent donc à trouver un équilibre entre la significativité (limiter le nombre de gènes sélectionnés à tort) et la puissance (limiter le nombre de gènes rejetés alors qu'ils sont bel et bien différentiellement exprimés) des tests pratiqués.

[6] Le FDR est l'espérance de la proportion de faux positifs parmi tous les tests significatifs

2.3.2.3 Approches paramétriques et non paramétriques

D'une manière générale, les différentes stratégies de test évoquées jusqu'ici génèrent des listes de gènes distinctes. Cela s'explique aisément par le fait que chacune d'entre elles fait un certain nombre d'hypothèses quant à la distribution des données et met l'accent sur différents aspects de ces dernières. Typiquement, de nombreux tests font l'hypothèse que les niveaux d'expression des gènes sont distribués selon une loi normale et ont une variance identique. Le plus souvent, on suppose également qu'ils sont indépendants (ce qui est en totale contradiction avec une conception systémique du fonctionnement cellulaire). Quelle que soit l'approche utilisée, il ne faut pas perdre de vue le fait que la liste de gènes obtenue à l'issue d'une expérience de puces à ADN ne constitue qu'une hypothèse. L'interprétation ou l'utilisation de cette liste est fortement dépendante des méthodes employées pour la générer.

2.3.2.4 Les listes de gènes différentiellement exprimés en discussion

En règle général seule une fraction des gènes présente des différences d'expression statistiquement significatives entre deux conditions expérimentales. Cela peut s'expliquer de différentes manières :

- dans des cellules ou des tissus distincts, un sous-ensemble de gènes sont exprimés essentiellement du fait de la différentiation cellulaire. Par conséquent, de tels gènes ne présentent théoriquement pas de différence dès lors que l'on compare les transcriptomes de cellules appartenant au même type ou au même tissu. Plus globalement, on s'attend à ce que seule une portion des gènes impliqués dans la différentiation cellulaire diffère entre deux types cellulaires ;
- un nombre conséquent de gènes codent pour des protéines qui sont nécessaires à la vie cellulaire. Toute altération sérieuse de leur expression serait donc létale ;
- comme nous l'avons expliqué dans la section précédente, la cellule dispose de nombreux moyens, autres que la régulation de la transcription, pour moduler la production des protéines : régulations traductionnelle ou post-traductionnelle notamment. Ceux-ci n'étant pas observables par la méthode des puces à ADN, le transcriptome des gènes concernés ne montrera aucune variation.

Il est difficile d'établir la fonction biologique de tous les gènes qui apparaissent différentiellement exprimés (et donc impliqués) dans un contexte physiologique ou pathologique donné. Cela requiert généralement une recherche bibliographique importante ainsi qu'un nombre non négligeable d'expériences complémentaires. Il est donc tentant de considérer des listes de taille restreinte afin de ne porter son attention que sur l'étude des gènes dont le différentiel d'expression est le plus tranché. Nous allons voir qu'il existe des méthodes d'analyse de données de puces à ADN permettant d'attribuer un rôle biologique à une liste de gènes différentiellement exprimés de manière automatique.

2.3.2.5 Sur-représentation de catégories fonctionnelles

Après avoir établi une liste de gènes dont l'expression montre une variation significative d'une condition expérimentale à une autre, il peut être intéressant d'étudier les catégories fonctionnelles auxquelles ces gènes appartiennent : c'est l'étape d'annotation fonctionnelle [7].
Afin de répondre aux besoins contradictoires d'augmenter la puissance de détection des expressions différentielles et de résoudre le problème posé par l'interprétation d'une longue liste de gènes

[7] L'annotation est le processus visant à identifier la protéine codée par un gène ainsi que la fonction remplie par celle-ci au sein du tissu étudié.

différentiellement exprimés, on recourt fréquemment à des tests d'enrichissement d'ensembles de gènes (*Gene Set Enrichment Analysis* ou GSEA [STM+05]). Il s'agit de déterminer si telle ou telle catégorie fonctionnelle de gènes apparaît de manière biaisée (sur- ou sous-représentée) dans la liste. Ces classes fonctionnelles sont généralement extraites à partir de *Gene Ontology* (GO). Il s'agit d'un moyen de décrire les produits des gènes en fonction du processus biologique ou de la fonction moléculaire qui leur est associé, indépendamment de l'espèce considérée [BRCA00]. Plus précisément, GO présente les différentes fonctions d'une protéine de manière hiérarchique, en partant de catégories très vastes pour arriver à des niveaux de description beaucoup plus précis. Par exemple, en termes de processus biologique, *ID2* (pour *inhibitor of DNA binding* ou inhibiteur de la fixation à l'ADN) est un gène du développement. En termes de fonction moléculaire, c'est un répresseur de l'activité transcriptionnelle.
De nouveau se pose la question de savoir si des ensembles de gènes sélectionnés au hasard n'induisent pas une sur- ou sous-représentation d'une catégorie fonctionnelle. Pour y répondre, on utilisera des tests statistiques permettant d'estimer la probabilité pour que la mise en avant d'une catégorie fonctionnelle par les gènes sélectionnés soit due au hasard. On dira alors qu'une catégorie fonctionnelle est significativement appauvrie ou enrichie parmi les gènes sélectionnés.

2.3.2.6 L'analyse différentielle en question

Supposons qu'à l'issue d'une expérience de puce à ADN comparant les tissus d'individus sains et malades, on identifie une liste de gènes différentiellement exprimés dont un sous-ensemble est clairement associé à une catégorie fonctionnelle de GO telle que le métabolisme des acides aminés. On pourra alors formuler l'hypothèse que cette pathologie altère la capacité des cellules à produire des acides aminés correctement. Évidemment, cette interprétation doit être manipulée avec la plus grande précaution. Plusieurs éléments nous poussent à relativiser une telle hypotèse.

Tous les niveaux de régulation ne sont pas pris en compte Même s'il est vrai qu'un nombre significatif de gènes régulés est impliqué dans le métabolisme des acides aminés, d'autres régulations opérant au niveau des protéines et non des ARN(m) peuvent survenir sans que l'on puisse les détecter. Par conséquent, d'autres fonctions moléculaires et cellulaires peuvent être modulées dans notre expérience sans que l'on soit en mesure de rendre compte d'une sur- ou sous-représentation de ces dernières parmi les gènes dont seules les concentrations en ARNm sont significativement modulées.

La variation du niveau d'expression d'un gène n'implique pas une perte ou un gain de fonction Dans le même ordre d'idées, une protéine peut conserver son efficacité (et donc sa fonction biologique) même si sa concentration chute de manière importante. Il arrive en effet que de très petites quantités d'une protéine soient suffisantes pour assurer une réaction chimique. Des gènes apparaissant comme régulés dans une situation particulière peuvent donc en réalité conserver le même effet au niveau cellulaire.

Les gènes peuvent avoir plusieurs fonctions N'oublions pas que la caractérisation fonctionnelle d'un gène n'est pas univoque. Les gènes peuvent intervenir à plusieurs niveaux dans une cellule. Par conséquent, certains des gènes sélectionnés peuvent en réalité agir de concert à un autre niveau ou dans une autre fonction que celle qui apparaît comme sur-représentée.

On observe une population hétérogène de cellules Enfin, rappelons que nous observons les profils d'expression au niveau d'une population de cellules qui peuvent aussi ne pas faire toutes exactement la même chose. En effet, un échantillon de cellules prélevé sur un patient peut contenir différents types de cellules. Du reste, indépendamment du type cellulaire auquel elles appartiennent, ces cellules ne sont pas dans le même état : par exemple, elles ne sont pas synchronisées et leurs gènes ne s'expriment pas de la même manière selon la phase du cycle cellulaire dans laquelle elles se trouvent. Dans ces conditions, la variation de niveau d'expression d'un certain nombre de gènes entre deux conditions expérimentales peut être imputée, dans une certaine mesure, à la composition même des échantillons utilisés.

Conclusions préliminaires D'une manière générale, les méthodes abordées jusqu'à présent sont faites pour répondre à des questions fermées que l'on résume par des hypothèses nulles : « les gènes présentent-ils le même niveau d'expression dans les différentes conditions expérimentales ? » ou « les catégories fonctionnelles sont-elles identiquement représentées parmi les gènes régulés ? ». Maintenant, nous allons essayer de nous poser des questions plus ouvertes concernant les systèmes de régulation pilotant les fonctions cellulaires. Pour cela, d'autres approches statistiques que les tests d'hypothèses peuvent être utilisées. Nous allons d'abord évoquer les méthodes de classification avant de nous intéresser dans le chapitre suivant à l'apprentissage de modèles de réseaux de régulation.

2.3.3 Classification des données de puces à ADN

Un processus de classification supervisée implique de placer des objets (en l'occurrence des gènes) dans des groupes (ou classes) préexistants. La classification non supervisée construit un ensemble de groupes dans lesquels les objets peuvent être rangés de manière optimale (au sens d'un critère qui reste à définir). Cette approche est intensivement utilisée en analyse de puces à ADN.

2.3.3.1 La classification supervisée

Parfois, les biologistes réalisent une expérience de puces à ADN afin d'infirmer ou de confirmer une hypothèse d'ordre biologique ou clinique. Il s'agit alors de réaliser une prédiction du type : étant donné le profil d'expression des gènes d'un patient, doit-on s'attendre à ce qu'il réponde au traitement proposé ? Cette prédiction de classe (ce terme n'ayant rien à voir avec les classes fonctionnelles issues de GO évoquées précédemment) repose sur des méthodes de classifications supervisées.
Le but de la classification supervisée est d'obtenir une fonction ou une règle qui utilise les données d'expression pour prédire la classe d'une observation (par exemple, le patient répond ou non à la thérapie testée). Un algorithme se charge de trouver la règle permettant de classer au mieux un ensemble de cas étiquetés (appelé *base d'apprentissage*), c'est-à-dire dont la classe est connue à l'avance. Dans notre exemple, chaque cas correspondra au profil d'expression d'un patient donné. L'étiquetage de ces cas revient à savoir si chaque patient correspondant est ou non sensible au traitement proposé.
Les méthodes de classification supervisée sont sujettes au sur-apprentissage* (bien que cela soit également vrai des méthodes de classification non supervisées). Afin d'estimer les performances de la règle apprise par l'algorithme sur un nouvel échantillon de test, on pratique généralement une validation croisée : cette règle est utilisée pour classer des données qui sont complètement indépendantes de celles utilisées pour son apprentissage. Bien sûr, une validation

croisée performante nécessite une taille d'échantillon adéquate. Des méthodes pour estimer les tailles d'échantillons pour les études de classification supervisée ont été développées [MTR+03, HSSS02].

2.3.3.2 La classification non supervisée

La plupart du temps, l'étude des profils d'expression se fait « en aveugle », aucune connaissance solide ne permettant de préjuger des résultats obtenus à l'issue de l'analyse (par exemple, pour les études de cinétique). Les puces à ADN constituent alors un outil d'investigation dont on espère qu'il fournira de nouvelles hypothèses candidates à tester. On réalise alors des expériences de détection de classes [Chu02] au moyen de méthodes de classification non supervisées.
L'objet de ces approches est de regrouper des gènes présentant les mêmes profils d'expression au sens d'un critère de similarité dépendant de l'algorithme. L'hypothèse invoquée pour justifier de tels regroupements est que des gènes présentant des profils d'expression similaires sont probablement co-régulés : par exemple, les mêmes facteurs de transcription moduleraient l'activité transcriptionnelle de ces gènes. Souvent, cette hypothèse est étendue à l'idée que des gènes co-régulés ont de fortes chances d'appartenir aux mêmes catégories fonctionnelles. Il est alors possible de mener une démarche dite « fautive par association » qui permet de donner une fonction probable à un gène inconnu dans la mesure où il est regroupé avec des gènes connus qui partagent la même fonction. Cela peut permettre par exemple de déterminer le type de fonctions moléculaires altérées par un toxique, afin de mieux saisir le mode d'action de ce dernier et de guider la recherche d'un remède. Le regroupement de gènes co-régulés permet donc de faire émerger une structure sous-jacente aux données.

Bien qu'elle constitue une approche séduisante, la classification apporte assez peu d'information quant aux mécanismes régissant les systèmes de régulation. Elle ne permet pas de caractériser les interactions régulatrices ni même de savoir si deux gènes appartenant au même groupe ont un régulateur commun ou si l'un régule l'autre.

Exemple 2.2
Des profils d'expression similaires pour deux gènes **a** *et* **b** *peuvent s'expliquer de trois manières distinctes. Soit* **a** *régule* **b**, *soit* **b** *régule* **a**, *soit* **a** *et* **b** *sont régulés par une tierce molécule. Dans tous les cas, les niveaux d'expression de* **a** *et* **b** *augmentent et diminuent de concert. Cela ne signifie pas que ces valeurs varient simultanément car si* **a** *régule* **b** *on est en droit de s'attendre à un délai entre l'augmentation du niveau de* **a** *et celle de* **b**. *Toutefois, dans la pratique, les puces à ADN ne permettent pas d'obtenir une résolution temporelle pour trancher. En effet, le laps de temps séparant deux mesures est généralement de l'ordre de l'heure, c'est-à-dire très supérieur au temps nécessaire pour réaliser les réactions biochimiques aboutissant à la régulation.*

Bien sûr, ces hypothèses sont largement discutables et les classes produites sont loin de structurer les données de manière suffisante pour permettre une véritable interprétation biologique en l'état.

La classification non supervisée est l'une des méthodes les plus populaires pour l'analyse de données de puces à ADN. Cette popularité peut s'expliquer par sa souplesse (aucune conjecture relative aux données n'est requise) et le fait que les chercheurs sont certains d'obtenir un regroupement des gènes indépendamment de la taille de l'échantillon, de la qualité des données et du dessin expérimental. On peut toutefois s'interroger sur la pertinence de cette approche : les regroupement réalisés sur des échantillons de taille restreinte (typiquement inférieurs

à 50) n'étant généralement pas reproductibles [GPS+05]. La reproductibilité des résultats engendrés par les méthodes de classification non supervisée doit donc être mesurée. En effet, les méthodes standards de classification ne fournissent aucune information quant à la propension des résultats à refléter un schéma existant au sein de la population ou bien de simples variations d'échantillonnage [8]. Des techniques de ré-échantillonnage peuvent être utilisées afin d'évaluer la reproductibilité de cette classification [KC01a, ZZ00, TW05] : comme précédemment, on étudie la variabilité des résultats produits par l'algorithme de classification lorsqu'il est appliqué à plusieurs jeux de données artificiels obtenus par ré-échantillonnnage des données réelles.

Bien que la plupart des travaux de classification se soient appuyés sur des méthodes telles que l'algorithme des k-moyennes ou les algorithmes de classification hiérarchique, de nouvelles approches plus élaborées se sont imposées. C'est notamment le cas du *biclustering* [RBB06] ou du *clustering* spectral [HKK07]. L'intérêt du biclustering vient du fait qu'il permet de regrouper des gènes présentant des profils d'expression similaires dans un *sous-ensemble de conditions expérimentales*, là où les méthodes antérieures cherchaient à identifier des profils d'expression similaires sur l'ensemble des observations disponibles. Cette dernière approche semble trop restrictive car lorsque l'on étudie des profils d'expression obtenus dans des conditions expérimentales extrêmement variées, on s'attend à ce que des gènes ne soient co-régulés que dans certaines d'entre elles.
D'une manière générale, une grande variété d'algorithmes de classification ont été utilisés pour regrouper des profils d'expression temporels et mettre en évidence des sous-ensembles de gènes co-régulés [BDY99, ESBB98, MCA+98]. Ces méthodes ont également été utilisées pour reconstruire des réseaux de régulation génétique à partir de données d'expression [DLS00].

Dans le chapitre suivant, nous allons présenter plus en détails la problématique de reconstruction des réseaux de régulation génétique, en présentant les principaux formalismes mathématiques permettant de modéliser ces réseaux et en discutant les méthodes permettant de les apprendre automatiquement à partir de données de profil d'expression.

[8] En statistique, la variation d'échantillonnage est la variabilité qui survient au sein d'échantillons aléatoires issus de la même population et qui est due uniquement au processus d'échantillonnage aléatoire.

Deuxième partie

Apprentissage des réseaux de régulation génétique

La modélisation est une phase fondamentale dans l'étude des réseaux de régulation génétique. Le première question que nous devons nous poser est : « que peut on attendre d'un modèle ? ».

Historiquement, la modélisation a surtout été un outil de représentation et d'investigation des réseaux de régulation génétique. De nombreux travaux ont étudié la capacité de divers formalismes mathématiques à décrire de manière *précise* les mécanismes régulatoires complexes présents dans la cellule. Les mathématiciens et les informaticiens se sont tous particulièrement intéressé à l'utilisation de modèles pour étudier la dynamique des réseaux de régulation. Cet aspect est souvent considéré comme crucial car il doit permettre de caractériser des propriétés dynamiques pouvant expliquer des phénomènes biologiques. Par exemple, un système de régulation multi-stationnaire peut expliquer des phénomènes de différentiations cellulaires alors qu'un système homéostatique peut rendre compte des processus vitaux nécessitant une forte stabilité. Toutefois, pour être à même d'interpréter un modèle, il est faut être en mesure de le construire. La plupart des modèles complexes utilisés pour étudier les réseaux de régulation ont été construits « à la main », par des experts, à partir de leurs connaissances et d'informations tirées de la littérature. Cela explique que ces travaux de modélisation aient surtout porté sur un petit nombre de situations biologiques, bien caractérisées au niveau moléculaire, appartenant souvent à des organismes modèles tels que la Drosophile (ou mouche du vinaigre) ou Saccharomyces cerevisiae (également appelée levure du boulanger).

Au lieu d'exploiter des modèles construits à partir de connaissances préétablies, nous souhaitons apprendre automatiquement des modèles à partir de données expérimentales. L'*apprentissage automatique de modèle* doit nous permettre d'interpréter des observations expérimentales pour extraire de nouvelles connaissances sur la structure des réseaux de régulation. Idéalement, nous aimerions que les modèles (souvent très puissants) construits par les experts puissent être appris automatiquement. Cependant, dans les faits, c'est rarement le cas. Le formalisme choisi pour une tâche d'apprentissage est fortement déterminé par la nature des données disponibles. Nous avons vu dans le chapitre précédent que l'utilisation des puces à ADN est une approche privilégiée pour étudier le comportement global d'une cellule. L'exploitation des données qui en sont issues est l'une des approches les plus prometteuses pour comprendre les réseaux de régulation. Tous les modèles ne sont cependant pas adaptés à la représentation de ce type de données. Surtout, certains modèles plus que d'autres offrent des attraits en termes d'apprentissage. Certains modèles probabilistes par exemple, permettent de tirer parti d'un grand nombre de résultats théoriques facilitant l'exploitation de données bruitées, parfois incomplètes, produites par des phénomènes complexes.

Dans ce chapitre nous allons donc nous efforcer de mettre en relief ces différents aspects. Dans un premier temps, nous présenterons quelques uns des formalismes mathématiques les plus utilisés pour étudier des réseaux de régulation dont la structure est déjà largement connue. Ces modèles sont présentés en fonction de la quantité et de la précision des informations qu'ils représentent. Nous commencerons par les équations différentielles avant d'aborder les modèles logiques (réseaux booléens et réseaux de René Thomas) en discutant à chaque fois l'adéquation du formalisme au système biologique représenté. À chaque fois, nous discuterons les mérites de ces formalismes en termes d'apprentissage, sans pour autant détailler les techniques d'apprentissage utilisées dans la mesure où elles s'éloignent notablement de notre travail.

Dans un second temps, nous nous intéresserons aux modèles à base de graphe qui sont à la fois les plus simples et les plus utilisés pour représenter les réseaux de régulation biologique dans les bases de connaissances. Nous détaillerons ensuite l'apprentissage des graphes non orientés au moyen de méthodes statistiques permettant d'exploiter efficacement les données de profils d'expression. Nous commencerons par les réseaux de co-expressions pour finir par les modèles graphiques Gaussiens. Ces derniers appartiennent à la famille des modèles graphiques non orientés.

Il s'agit d'une sous-famille des modèles graphiques qui nous semblent constituer un cadre pertinent pour l'apprentissage automatique des réseaux de régulation. Plus particulièrement, nous avons décidé de travailler sur l'apprentissage de réseaux Bayésiens qui sont des modèles graphiques orientés. Le second chapitre de cette partie sera donc consacré à l'apprentissage de réseaux Bayésiens, que nous présenterons plus en détails.

Chapitre 3

Modélisation et reconstruction des réseaux de régulation génétique

Dans ce chapitre, nous abordons la problématique de l'apprentissage de modèle et nous présentons les données susceptibles d'être utilisées pour apprendre des réseaux de régulations génétiques. Nous introduisons ensuite différents formalismes mathématiques plus particulièrement adaptés à la modélisation et à l'étude dynamique des réseaux de régulations. Après avoir discuté de leur qualité en terme d'apprentissage, nous nous intéressons plus particulièrement aux modèles à base de graphe. Par le biais de l'apprentissage dans les réseaux d'association, nous présentons les modèles graphiques Gaussiens et introduisons la question de l'apprentissage de structure dans les modèle graphiques orientés. Ces derniers seront traités en détail dans le chapitre suivant.

3.1 Des modèles pour représenter, analyser ou apprendre les réseaux de régulation

Classiquement, le choix du formalisme mathématique utilisé pour représenter un système de régulation est conditionné par les deux questions suivantes : quel est le niveau de description souhaité par le modélisateur et quel usage entend-il faire du modèle à sa disposition ?

Représenter les réseaux de régulation Le formalisme mathématique utilisé pour modéliser un réseau de régulation peut être choisi en fonction de sa capacité à représenter des informations plus ou moins nombreuses et précises quant aux mécanismes moléculaires régissant le système régulatoire. Éventuellement, ce choix peut être guidé par les vertus explicatives des modèles. Les utilisateurs finaux étant fréquemment des chercheurs en sciences de la vie, l'utilisation de représentations graphiques aisément interprétables peut être un avantage. En se fondant sur ces considérations, il est possible de distinguer deux types de modèles.

Les modèles quantitatifs Si l'on souhaite représenter l'ensemble des réactions biochimiques impliquées dans la régulation de l'expression des gènes, il est nécessaire de décrire de manière précise l'interaction de chaque protéine régulatrice avec l'ADN du gène cible, son ARN(m), ou toute protéine participant à la chaîne de biosynthèse du produit final du gène cible. On s'intéressera donc à des modèles très riches tels que les équations différentielles.

Les modèles qualitatifs Si on se satisfait d'une représentation plus abstraite et synthétique de la régulation génétique, il est possible de ne s'intéresser qu'aux liens de causalité entre l'expression des gènes régulateurs et celle des gènes régulés. On se tournera alors vers des

modèles qualitatifs tels que les réseaux booléens ou les réseaux multivalués (modèles de René Thomas). S'il n'est pas utile de préciser la nature des interactions régulatrices, il est également possible d'utiliser des graphes (orientés ou non orientés). Dans ce cas, seule l'existence des relations entre molécules biologiques est représentée.

Analyser les propriétés des réseaux de régulation Les modèles que nous allons présenter ne se différencient pas seulement par leurs qualités descriptives. Si l'on veut analyser les propriétés dynamiques d'un réseau de régulation, simuler et prédire son comportement dans le temps, il est nécessaire de recourir à des modèles élaborés tels que les équations différentielles, les réseaux booléens, ou les réseaux multivalués de Thomas. Si l'on souhaite analyser ses caractéristiques topologiques, c'est-à-dire n'étudier que la structure de ce dernier indépendamment de la nature des interactions régulatrices qui le composent, l'utilisation des graphes est suffisante.

3.1.1 Problématique de la reconstruction de réseaux de régulation

La reconstruction automatique de modèles à partir de données expérimentales revêt un intérêt fondamental en biologie des systèmes. En effet, la construction d'un modèle requiert une masse conséquente de connaissances dont le modélisateur ne dispose que très rarement. Dans le même temps, la promesse d'une disponibilité croissante de données permettant de mesurer différents états d'un système de régulation a favorisé le développement de méthodes d'analyse tirant parti du caractère global de ces mesures. À ce titre, l'utilisation de la classification pour identifier des gènes potentiellement co-régulés est emblématique. L'idée de remplacer le processus de construction de modèles (réalisé par des experts) qui est à la fois long, difficile et coûteux par un processus d'apprentissage automatique de modèles à partir de données d'expression de gènes s'est donc fortement développée au cours des années passées.

Différents formalismes mathématiques permettent de reconstruire automatiquement un réseau de régulation génétique à partir de données expérimentales [MS07]. En fonction du résultat souhaité il est possible d'envisager l'apprentissage de modèles qualitatifs ou quantitatifs, statiques ou dynamiques. Cependant, le choix du formalisme employé pour l'apprentissage dépend essentiellement de la nature des observations dont on dispose. Il varie selon que les données reflètent les concentrations des ARN, des protéines et/ou des métabolites du système. Il varie également selon que ces données sont dynamiques (les mesures sont effectuées dans le temps) ou statiques (les mesures sont effectuées dans différentes conditions expérimentales, indépendamment du temps), discrètes ou continues. Enfin, ce choix dépend la nature du système à identifier et de l'approche théorique que l'on souhaite mettre en œuvre pour y parvenir. Comme nous allons le voir, en fonction de la famille de modèles sélectionnée, il est plus ou moins facile d'exploiter des données bruitées pour apprendre la structure d'un réseau reposant sur des influences régulatrices complexes et non déterministes.

3.1.2 Les données expérimentales à traiter

Afin d'éclairer les critiques formulées à l'encontre de certains modèles présentés par la suite, nous allons présenter les hypothèses que nous avons faites concernant les données d'apprentissage.

Des données d'expression de gènes La première question qui se pose est de savoir quelles sont les espèces moléculaires mesurées dans les données expérimentales. Un modèle est d'autant plus précis que les informations utilisées pour le construire sont complètes. Malheureusement, il n'existe pas de méthode à grande échelle permettant d'étudier le métabolome*. Il est possible de mesurer le protéome*, cependant les méthodes utilisées (électrophorèse bidimensionnelle ou

spectrométrie de masse), bien que très résolutives, sont particulièrement coûteuses et difficiles à mettre en œuvre. De plus, l'analyse de la masse de données brutes qu'elles génèrent est un problème encore largement ouvert. Actuellement, seule l'étude du transcriptome* bénéficie de méthodes et d'outils permettant de générer des données exploitables en quantité raisonnable.

Bien qu'elle n'offre qu'une vision partielle des systèmes de régulation, l'étude du transcriptome peut être utilisée pour déterminer les mécanismes de régulation de l'expression des gènes. La technique la plus utilisée pour mesurer simultanément le niveau d'expression d'un grand nombre de types différents d'ARN messagers est celle de la puce à ADN présentée en section 2.2. Notre hypothèse de base concernant les données d'apprentissage est donc que la reconstruction de modèles de régulation se fonde essentiellement sur des données de profils d'expression. À terme, il va de soi que des données plus complètes seront nécessaires à l'élaboration de modèles plus réalistes de la régulation génétique.

Des données discrètes Le choix entre un modèle discret ou continu est fortement déterminé par la nature des données disponibles. Les données brutes de puces à ADN étant des intensités de fluorescence, elles sont naturellement continues. Il est cependant assez rare de les conserver en l'état. Comme nous l'avons souligné dans le chapitre introductif, il est courant de réaliser une analyse différentielle. Bien que l'on puisse exprimer un différentiel d'expression au moyen d'un ratio d'intensité de fluorescence, l'utilisation de tests statistiques (nécessaires dès que l'on s'intéresse à des mesures sur le vivant) permettent souvent de caractériser l'expression d'un gène au moyen de valeurs discrètes telles que : induit, réprimé, ou non modulé (pour certaines conditions expérimentales et en référence à un cas contrôle). D'une manière générale il est courant de discrétiser les données de puces à ADN dans l'espoir de diminuer les effets des bruits (d'origine biologique ou expérimentale) inhérents à ce type de mesures. Bien que l'on puisse discuter du gain en termes de robustesse des données discrétisées par rapport aux données continues, les premières sont souvent préférées car elle sont souvent plus faciles à exploiter dans un cadre d'apprentissage. Il faut cependant garder à l'esprit que la discrétisation des données peut induire une perte d'information préjudiciable à l'extraction de connaissances.

Des données statiques Compte tenu des phénomènes physico-chimiques à l'œuvre dans la cellule, il semble naturel de tenir compte du temps dans un modèle. Les données cinétiques nécessaires à l'apprentissage de modèles dynamiques ne sont en fait disponibles que pour des organismes relativement simples, en général monocellulaires (*E. coli* ou *saccharomyces cerevisiae* [SSZ+98]). En outre, les échantillons susceptibles d'être utilisés sont pseudo-temporels dans la mesure où les observations sont en générales destructrices : pour acquérir une cinétique de 10 points, il est nécessaire de lancer 10 cultures synchronisées et d'utiliser une culture pour chaque prélèvement nécessaire à 1 point. Par ailleurs, le nombre de points d'une cinétique est assez faible en comparaison du nombre de mesures statiques effectuées, par exemple dans le cadre d'études cliniques sur des cohortes de plusieurs centaines de patients.
D'une manière générale, les expériences de puces à ADN sont plutôt utilisées pour comparer le transcriptome d'un organisme modèle dans des conditions expérimentales variées. Il s'agit par exemple de comparer les profils d'expression d'individus sains et d'individus malades à différents stades de leur pathologie. On étudie également les effets de différents toxiques (rayonnements ionisants ou métaux lourds) ou médicaments sur le transcriptome des patients en fonction des doses reçues. On parle alors de données de *perturbation*, ces dernières mesurant la réponse du système à un stimulus qui perturbe ponctuellement son fonctionnement.
Il peut être particulièrement avantageux de recourir à des perturbations ciblées, visant un gène en particulier, afin de produire des données *d'intervention*. Il est par exemple possible d'utiliser

des micro-ARN ou des ARN interférants (voir section 1.1 page 10) afin d'inhiber spécifiquement des gènes préalablement choisis. Les profils d'expression résultant permettent alors d'observer les effets de ces interventions sur l'expression des autres gènes. Par exemple, inhiber un gène X_k dans une cascade de régulation $X_i \rightarrow X_k \rightarrow X_j$ ($X_k \rightarrow X_j$ signifie « X_k régule X_j ») n'affectera que l'expression de X_j. Si X_k régule à la fois X_i et X_j, le fait d'intervenir sur X_k modifiera le comportement de ces deux cibles. Par contre, si $X_i \rightarrow X_k \leftarrow X_j$ alors ni l'expression de X_i, ni celle de X_j ne seront modifiées par une inhibition de X_k. Les interventions ciblées sont donc très utiles lorsque l'on souhaite comprendre les relations existant entre des gènes. Elles sont cependant plus difficiles à mettre en œuvre que des perturbations et impliquent que l'on ait déjà une idée précise des gènes que l'on souhaite étudier, et donc cibler.
Les techniques et les besoins évoluant, la quantité et la qualité des jeux de données cinétiques devraient augmenter avec le temps. Nous avons fait le choix de privilégier l'étude de formalismes et de méthodes permettant d'exploiter des données statiques du fait de leur plus grande disponibilité.

La variabilité des données Les données de puces à ADN (et les données en biologie en générale) sont des données présentant une variabilité importante. Elles sont tout d'abord caractérisées par un bruit de nature intrinsèque. En effet, ces données peuvent être vues comme des échantillons d'un processus aléatoire, les phénomènes biologiques observés via les mesures de profils d'expression étant de nature stochastique. Cela est d'autant plus problématique que le caractère stochastique des réseaux de régulation n'est pas seulement un aspect inhérent à la vie cellulaire.

Les données de puces à ADN présentent également un bruit de nature extrinsèque, s'expliquant essentiellement par des considérations d'ordre expérimental. Nous avons vu dans le chapitre précédent qu'une expérience de puce à ADN contient de nombreuses étapes. Chacune d'entre elles est une source potentielle de variabilité, depuis la fabrication même de la puce (dans le cas des puces spotées) dont les sondes peuvent être de qualité variable, jusqu'aux procédures d'extraction ou d'hybridation des ARNm. Il y a deux autres aspects qu'il faut également prendre en compte. Premièrement, ces méthodes ne s'appliquent jamais à une cellule unique mais à un ensemble de cellules, parfois mises en culture, parfois extraites d'un tissu. Les informations collectées caractérisent donc un comportement moyen d'un ensemble de cellules. Deuxièmement, lorsque les cellules utilisées sont obtenues à partir d'un tissu, elles n'appartiennent pas nécessairement toutes au même type cellulaire. En effet, les méthodes de tri cellulaire [1] produisent toujours des échantillons contenant une faible proportion de cellules « non désirées ». Par conséquent, on étudie le profil d'expression d'un ensemble de cellules qui n'est pas réellement homogène.

Dans ce qui suit, nous allons présenter certains formalismes mathématiques particulièrement populaires pour la modélisation des réseaux de régulation biologique. Ce panorama ne prétend pas à l'exhaustivité, certaines familles de modèles tels que les réseaux de Pétri n'étant pas abordées. Nous souhaitons insister sur la difficulté de faire correspondre la réalité des systèmes biologiques et les outils mathématiques à notre disposition en présentant des formalismes plus ou moins résolutifs : des plus précis (les équations différentielles) aux plus qualitatifs (les graphes). Nous nous intéresserons, à chaque fois que cela est possible, à la question de leur apprentissage en essayant de mettre en avant les difficultés soulevées par la nature de ces familles de modèles et par les données disponibles.

[1] Méthodes grâce auxquelles on ne retient que les cellules appartenant à un type cellulaire spécifique, au sein d'un échantillon de cellules distinctes.

3.2 Les modèles différentiels

Les modèles différentiels sont très probablement le formalisme le plus répandu pour la modélisation dynamique des systèmes biologiques [CHC99, CCNG+00, ASI02, YZO+06]. Bien qu'ils ne concernent pas directement nos travaux, il nous semble intéressant de s'y attarder car ils permettent d'illustrer la plupart des aspects inhérents à la modélisation des systèmes de régulation.
Les modèles différentiels permettent de représenter de manière très précise les mécanismes moléculaires sur lesquels s'appuient les phénomènes de régulation. Pour cela, ils représentent les concentrations des espèces moléculaires en jeu dans la cellule telles que les ARNm, les protéines et les métabolites, au moyen de variables (réelles positives) dépendant du temps. La variation de ces grandeurs est décrite par un système d'équations différentielles couplées. Lorsque l'on ne tient compte que des espèces moléculaires en présence et de leurs interactions, on a un système d'équations différentielles ordinaires (EDO). Chaque équation formalise l'évolution d'une variable de manière continue au cours du temps, en fonction des concentrations des molécules exerçant une influence régulatrice sur cette dernière. Plus précisément, une régulation génétique est modélisée par une équation *de taux de réaction* exprimant le taux de production d'un produit de gène (un ARNm ou une protéine) en fonction des concentrations des autres éléments du système impliqués dans sa production. Un exemple est représenté à la figure 3.1. Pour favoriser la lisibilité des figures, la dérivé d'une variable x_i par rapport au temps y est notée $\dot{x_i}$ au lieu de $\frac{dx_i}{dt}$ dans le texte. Les équations de taux de réaction sont de la forme :

$$\frac{dx_i}{dt} = f_i(\mathbf{x}), x_i \geq 0, 1 \leq i \leq n, \tag{3.1}$$

où x_i est la concentration de la molécule X_i, $\mathbf{x}$ est le vecteur des concentrations des espèces moléculaires impliquées dans la production de X_i (x_i pouvant elle-même en faire partie dans le cas d'une rétroaction) et f_i est une fonction généralement non linéaire.
Une fonction de régulation couramment employée pour la modélisation des réseaux de régulation génétique est la fonction de Hill (voir figure 3.2) :

$$h_i^+(x_j, \theta_{ij}, m) = \frac{x_j^m}{x_j^m + \theta_{ij}} \tag{3.2}$$

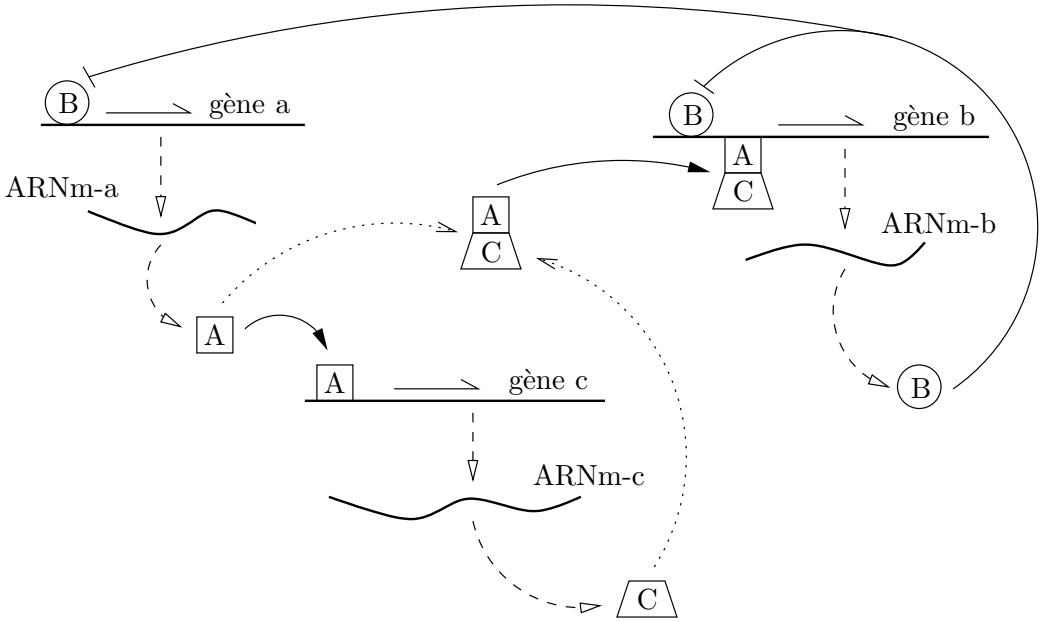
où $\theta_{ij} > 0$ est le seuil au-delà duquel l'influence de j sur i change de régime, et $m > 0$ défini l'amplitude du saut effectué par la concentration de i après franchissement du seuil θ_{ij}. Cette fonction prend ses valeurs dans l'intervalle $[0, 1]$ et croît strictement avec x_j. Il s'agit donc d'une fonction d'induction qui augmente le taux d'expression de i lorsque x_j augmente. Afin d'exprimer la situation opposée, à savoir une répression, la fonction de régulation $h_i^+(x_j, \theta_{ij}, m)$ est remplacée par $h_i^-(x_j, \theta_{ij}, m) = 1 - h_i^+(x_j, \theta_{ij}, m)$.

Outre les concentrations des molécules régulatrices, il est possible d'inclure l'effet de molécules extérieures sur la concentration de l'espèce i au sein du modèle.

$$\frac{dx_i}{dt} = f_i(\mathbf{x}, \mathbf{u}), x_i \geq 0, 1 \leq i \leq n \tag{3.3}$$

où u est la concentration d'une molécule extérieure au système, telle qu'une drogue dont on souhaite étudier les effets par exemple.

Il est également possible de raffiner le modèle par la prise en compte de la notion de délais correspondant au temps nécessaire pour transcrire un gène, traduire un ARNm (ces mécanismes



Concentrations en ARNm	Concentrations en protéines
$\dot{x}_a = k_{aB} \cdot r_a(\dot{x}_B) - \gamma_a \dot{x}_a$	$\dot{x}_A = k_{Aa}\dot{x}_a - \gamma_A \dot{x}_A$
$\dot{x}_b = k_{bABC} \cdot r_b(\dot{x}_A, \dot{x}_B, \dot{x}_C) - \gamma_b \dot{x}_b$	$\dot{x}_B = k_{Bb}\dot{x}_b - \gamma_B \dot{x}_B$
$\dot{x}_c = k_{cA} \cdot r_c(\dot{x}_A) - \gamma_c \dot{x}_c$	$\dot{x}_C = k_{Cc}\dot{x}_c - \gamma_C \dot{x}_C$

Fig. 3.1 – Un exemple de réseau de régulation biologique ainsi que le système d'équations différentielles correspondant. Les équations différentielles expriment la production des protéines et des ARNm codés par chacun des gènes du système de régulation. La variation de la concentration de chaque espèce moléculaire dépend d'un terme de production et d'un terme de dégradation. La dégradation d'une molécule X_i est proportionnelle à sa concentration et à un taux de dégradation γ_i. Le terme de production dépend d'une constante de taux $k_{i.}$. La production de chaque protéine est proportionnelle à la concentration de l'ARNm correspondant (car aucune régulation traductionnelle n'intervient). La production de chaque ARNm X_i dépend d'une fonction non linéaire r_i des concentrations des protéines régulatrices.

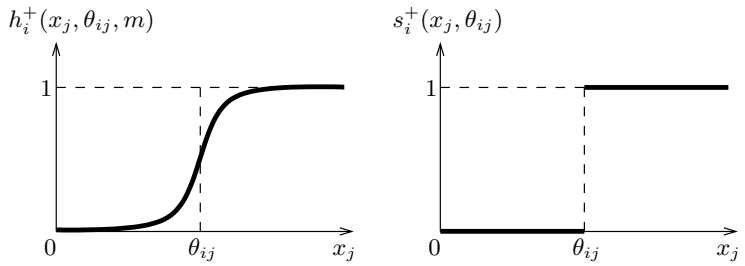


Fig. 3.2 – Fonction de Hill et fonction en escalier. (1) Une illustration de la fonction de Hill modélisant l'influence régulatrice de x_j sur x_i. Le paramètre de seuil θ_{ij} est propre à un régulateur et à sa cible. Le paramètre m est constant. (2) Une illustration de la fonction en escalier. La variable régulée x_i change d'état lorsque la variable régulatrice x_j atteint le seuil θ_{ij}. Cette fonction constitue une approximation de la fonction de Hill.

n'étant pas instantanés, ni synchronisés) ou bien pour permettre à l'une de ces molécules de diffuser d'un compartiment à l'autre au sein de la cellule :

$$\frac{dx_i}{dt} = f_i(x_1(t-\tau_{i1}), \ldots, x_n(t-\tau_{in})), x_i \geq 0, 1 \leq i \leq n, \tag{3.4}$$

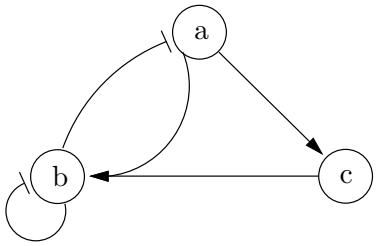
où $\tau_{i1}, \ldots, \tau_{in} > 0$ représentent les délais propres à chacune des molécules impliquées dans la production de i.

Du fait du caractère non linéaire des fonctions f_i, il n'est généralement pas possible de trouver des solutions analytiques aux équations du taux de réaction afin d'identifier leurs états stables ou des cycles limites. Il est courant de recourir à des techniques d'analyse numérique afin d'approcher la solution exacte de ces équations [But00]. De nombreux outils logiciels dédiés à la simulation de réseaux de régulation biologique décrits par des équations différentielles sont disponibles [Men93, GHS99, dJGHP03].

La simulation d'un réseau de régulation est souvent complétée par des techniques issues de l'étude des systèmes dynamiques, telles que les outils d'analyse de bifurcation visant à étudier la sensibilité des états stationnaires et des cycles limites aux valeurs des paramètres du modèle [Str00]. Ces approches ont par exemple été appliquées à l'étude d'un modèle numérique décrivant un cas très étudié de modification post-traductionnelle impliquée dans le contrôle de la mitose chez l'oocyte de Xenope [BT98]. D'une manière générale, l'analyse cinétique est réalisée sur des systèmes biologiques relativement simples et très bien caractérisés, chez des organismes modèles tels que la levure [NT95, CCNG$^+$00] ou le phage λ [RV90, MS95].

3.2.1 Les équations différentielles linéaires par morceaux

Lorsque des équations différentielles non linéaires ne peuvent être analysées mathématiquement, une approche alternative à l'analyse numérique consiste à simplifier le modèle que l'on souhaite étudier. Cette approche se justifie également par le fait que la construction d'un modèle cinétique complet requiert une connaissance détaillée des mécanismes décrits par ce modèle. Les informations en question étant rarement disponibles, il est souvent nécessaire d'utiliser une modélisation moins fine. Dans cette optique, une classe de modèles couramment employée est celle des équations différentielles linéaires par morceaux (EDLM) [GK73, MPO95, dJGHP03, JGH$^+$04].



$$\begin{aligned}
\dot{x}_a &= k_a + k_{aB} \cdot h^-(\dot{x}_B, \theta_{aB}, m) - \gamma_a \dot{x}_a \\
\dot{x}_b &= k_b + k_{bAC} \cdot h^+(\dot{x}_A, \theta_{bA}, m) \cdot h^+(\dot{x}_C, \theta_{bC}, m) \\
&\quad + k_{bB} \cdot h^-(\dot{x}_B, \theta_{bB}, m) - \gamma_b \dot{x}_b \\
\dot{x}_c &= k_c + k_{cA} \cdot h^+(\dot{x}_A, \theta_{cA}, m) - \gamma_c \dot{x}_c
\end{aligned}$$

Fig. 3.3 – Un réseau de régulation génétique modélisé par un système d'équations différentielles linéaires par morceaux.. Un réseau de régulation génétique représente les influences régulatrices entre gènes. La variation de la concentration de chaque espèce moléculaire dépend d'un terme de production et d'un terme de dégradation. La dégradation d'une molécule X_i est proportionnelle à sa concentration et à un taux de dégradation γ_i. Le terme de production est la somme du niveau d'expression basal k_i de la molécule et de termes de régulation dépendant chacun d'un ensemble de régulateurs J. Chacun de ces termes de régulation est le produit d'une constante de taux k_{iJ} et de fonctions de Hill dépendant de chacun des régulateurs $j \in J$. Les fonctions de Hill peuvent être remplacées par des fonctions discontinues en escalier de seuils θ_{ij}.

Ces dernières sont issues des équations de taux de réaction (3.1) ayant fait l'objet d'hypothèses simplificatrices.
On considère des équations de taux de réaction de la forme :

$$\frac{dx_i}{dt} = g_i(\mathbf{x}) - \gamma_i x_i, x_i \geq 0, 1 \leq i \leq n, \tag{3.5}$$

où x_i dénote la concentration cellulaire du produit du gène X_i, et $\gamma_i > 0$ est le taux de dégradation de x_i. Ce paramètre permet de modéliser la dégradation des molécules biologiques dans l'organisme. Celle-ci est particulièrement importante pour les ARN messagers qui sont des molécules très fragiles. Les fonctions g_i sont définies comme la somme de termes d'interactions correspondant aux régulations coopératives dans le réseau. Plus précisément, à chaque régulation coopérative d'un gène X_i par un ensemble de gènes régulateurs J, correspond un terme $k_{iJ} \prod_{j \in J} r_i(x_j)$, où r_i est une fonction de régulation (de Hill, le plus souvent) et k_{iJ} une constante spécifiant le niveau d'expression maximum de X_i en fonction de J. Ici, on modélise le fait que certaines molécules régulatrices concourent de façon complémentaire à la régulation. Typiquement, un assemblage moléculaire régule la transcription de certains gènes. Ses différentes composantes (un ensemble J de molécules) doivent être présentes en quantité suffisante pour que le polymère puisse être constitué afin de réguler X_i. Bien sûr, différents polymères peuvent réguler le gène cible indépendamment les uns des autres.

$$\frac{dx_i}{dt} = \sum_J k_{iJ} \prod_{j \in J} r_i(x_j) - \gamma_i x_i, x_i \geq 0, 1 \leq i \leq n, \tag{3.6}$$

avec $J \subset \mathbf{X}$ où $\mathbf{X} = \{X_1, \ldots, X_n\}$. Un exemple d'application de ce formalisme aux réseaux de régulation génétique est fourni à la figure 3.3. On remarquera que cet exemple reprend

le réseau de régulation biologique de la figure 3.1 et le simplifie en faisant abstraction des mécanismes moléculaires intervenant aux différentes étapes de la régulation. Par ailleurs, le caractère coopératif ou indépendant des régulations est explicité dans les formules.

Afin de simplifier l'analyse mathématique, les fonctions de Hill continues sont remplacées par des fonctions discontinues en escalier.

$$s_i^+(x_j, \theta_{ij}) = \begin{cases} 1, x_j > \theta_{ij} \\ 0, x_j < \theta_{ij} \end{cases}, \qquad \text{et} \qquad s_i^-(x_j, \theta_{ij}) = 1 - s_i^+(x_j, \theta_{ij}) \tag{3.7}$$

Cette approximation a été proposée depuis longtemps par de nombreux auteurs [Gla75, GK73, SF63, Tho73, WPY67]. Elle se justifie par le fonctionnement de l'expression des gènes, parfois similaire à celui d'un interrupteur : au-dessous d'une certaine concentration, le régulateur tend à avoir une influence très faible (parfois négligeable) sur sa cible, tandis qu'au-delà de cette concentration, l'influence du régulateur devient significative et atteint rapidement un niveau maximal.
Les équations de taux d'expression qui en résultent sont des équations différentielles linéaires par morceaux de la forme :

$$\frac{dx_i}{dt} = b_i(\mathbf{x}) - \gamma_i x_i, x_i \geq 0, 1 \leq i \leq n, \tag{3.8}$$

où b_i est une fonction constante par morceaux. Plus précisément, b_i est une somme de produits de fonctions à seuil pondérées par une constante de taux. Les EDLM ont été très étudiées en biologie des systèmes grâce notamment à leurs propriétés mathématiques qui favorisent grandement l'analyse dynamique [MPO95, HSKG99, dJP08]. La question qui se pose est de savoir quelle est la quantité d'information perdue, à cause de l'approximation linéaire par morceaux du système non linéaire original. Des études fondées sur des simulations numériques [GK72, GK73, PMO95, PMO98] ont montré que dans un certain nombre de cas, les comportements dynamiques obtenus à partir de ces deux classes de modèles ne présentaient pas de différences qualitatives significatives. Les EDLM sont une version simplifiée des EDO et à ce titre elles ne sauraient offrir la même richesse de description et de comportement, cependant la question qui importe est de savoir si un modèle plus qualitatif peut capturer les propriétés pertinentes du système, nécessaires à sa compréhension. Surtout, une telle simplification constitue un atout en terme d'apprentissage. En effet, comme nous le verrons par la suite, les paramètres d'un modèle sont d'autant plus difficile à estimer que ce modèle est complexe. D'une manière générale, le principe du rasoir d'Occam nous suggère de choisir le modèle le plus simple parmi les modèle expliquant le mieux le comportement du système modélisé.
Notons également qu'il est possible de simplifier encore ce modèle en utilisant des fonctions logiques pour modéliser l'induction ou la répression d'un gène en fonction de l'état de ses régulateurs (actifs ou inactifs selon que leur concentration dépasse ou non un certain seuil). La différence essentielle avec les réseaux booléens que nous découvrirons dans la section suivante vient du fait que l'évolution de la concentration du produit du gène cible est continue dans le temps [KEG03].

3.2.2 Les équations différentielles stochastiques

Nous l'avons vu, les EDO permettent de décrire les phénomènes de régulation génétique de manière très détaillée. Elles autorisent la prise en compte d'interactions moléculaires individuelles telles que la fixation d'un facteur de transcription à l'ADN ou l'association de plusieurs protéines pour former un complexe. Cependant, un tel niveau de détail est souvent inutile du fait de l'absence d'informations moléculaires suffisamment précises pour construire ce modèle. Plus

généralement, un certain nombre d'hypothèses sous-jacentes à l'utilisation de ce formalisme se révèlent fausses lorsque l'on travaille au niveau moléculaire. Les EDO présupposent que les concentrations des substances d'intérêt varient de manière continue et déterministe, or ces deux points sont discutables lorsque l'on s'intéresse à la régulation génétique [Gil77, Sza99]. Par exemple, le temps nécessaire au déroulement d'un certain nombre de mécanismes moléculaires peut dépendre du temps de diffusion des réactifs à travers la cellule. Le temps de diffusion peut lui-même varier en fonction du volume de la cellule et de la localisation des réactifs en son sein. D'une manière générale, on peut légitimement douter que le délai entre le début et la fin de la transcription d'un gène, lorsqu'il est pris en compte, soit constant. L'évolution déterministe du système et, d'une manière générale, la « synchronisation » des différents processus modélisés, doit donc être remise en cause. Si des fluctuations surviennent dans le déroulement temporel des phénomènes étudiés, deux systèmes ayant le même état initial peuvent atteindre des états stables distincts.
Il a donc été proposé d'adopter une modélisation fondée sur une représentation discrète et stochastique de phénomènes de régulation [ARM98, Gil77, MA97].
Soient des quantités discrètes de molécules, notées $\mathbf{x}$, correspondant à des variables d'état. On considère la distribution de probabilité jointe $P(\mathbf{x}, t)$ exprimant la probabilité de voir au sein d'une cellule une quantité $x_i \in \mathbf{x}$ de molécules $X_i \in \mathbf{X}$ au temps t. L'évolution temporelle de la fonction $P(\mathbf{x}, t)$ peut alors être spécifiée comme suit :

$$P(\mathbf{x}, t + \Delta t) = P(\mathbf{x}, t) \left(1 - \sum_{j=1}^{m} \alpha_j \Delta t \right) + \sum_{j=1}^{m} \beta_j \Delta t \tag{3.9}$$

où m est le nombre de réactions caractérisant le système, $\alpha_j \Delta t$ la probabilité que la réaction j se produise durant l'intervalle de temps $[t, t + \Delta t]$ sachant que le système se trouve dans l'état $\mathbf{x}$ au temps t, et $\beta_j \Delta t$ la probabilité de voir la réaction j ramener le système à l'état $\mathbf{x}$ durant $[t, t + \Delta t]$ [Gil77, Gil92]. En ré-arrangeant cette équation afin de prendre sa limite lorsque Δt tend vers 0, on obtient l'équation maîtresse [Kam07] :

$$\frac{dP(\mathbf{x}, t)}{dt} = \sum_{j=1}^{m} (\beta_j - \alpha_j P(\mathbf{x}, t)) \tag{3.10}$$

On constate que les équations du taux de réaction (3.1) introduites précédemment déterminent la façon dont l'état du système évolue avec le temps, alors que l'équation ci-dessus rend compte de la probabilité de voir le système dans un certain état à un instant donné.

Bien que l'équation maîtresse fournisse une représentation plus conforme aux processus stochastiques gouvernant la dynamique des systèmes de régulation, elle est aussi plus difficile à analyser mathématiquement que sa contrepartie déterministe. Différentes alternatives peuvent être envisagées afin d'étudier ces modèles. L'une des principales méthodes utilisées consiste à simuler directement l'évolution stochastique du système au cours du temps en se fondant sur l'approche de simulation stochastique développée par Gillespie [Gil77]. Pour être concis, l'algorithme de simulation stochastique est un processus itératif qui détermine à chaque étape la nature de la prochaine réaction chimique et le laps de temps au bout duquel celle-ci doit se réaliser sachant que le système est dans l'état $\mathbf{x}$ au temps t. L'état du système est alors mis à jour en fonction de cette réaction. Contrairement à l'équation maîtresse qui rend compte d'un comportement moyen, cet algorithme propose une information sur un comportement particulier du système. L'idée est qu'à l'issue d'un nombre suffisant de simulations stochastiques, la distribution de $\mathbf{x}$ au temps t soit proche de celle décrite par l'équation maîtresse.

Bien que les prédictions réalisées grâce à cette approche se soient avérées cohérentes avec des observations expérimentales, elles demeurent sensibles vis-à-vis de la variation d'un certain nombre de paramètres [GB98]. La modélisation et la simulation stochastique constituent donc un moyen d'approcher de manière plus réaliste les aspects moléculaires de la régulation [SKPG05, TCM04]. Malgré tout, elle demeure fortement dépendante de la disponibilité de connaissances très fines quant au système modélisé et de l'estimation des paramètres utilisés.

3.2.3 Apprentissage de modèles différentiels

D'une manière générale, la richesse des modèles différentiels est peu exploitée dans le cadre de l'apprentissage. La plupart du temps, les observations disponibles ne rendent pas compte des concentrations des protéines présentes dans la cellule. Lorsqu'on ne mesure que des concentrations en ARNm, disposer de modèles capables de représenter le détail les réactions biochimiques concourant à la régulation est inutile. La prise en compte de délais dans les réactions biochimiques [CHC99] est également discutable. En effet, les puces à ADN ne permettent pas de mesurer les concentrations en ARN à des intervalles de temps suffisamment courts pour capturer les délais des mécanismes de régulation. Enfin, il est difficile de prendre en compte la nature stochastique des réseaux de régulation et le caractère bruité des observations, les équations différentielles stochastiques étant trop complexes pour être traitées efficacement [RHCC07].

Différents travaux ont essayé de tirer profit de données d'expression pour estimer les paramètres d'un modèle différentiel, le plus souvent par la méthode des moindres carrés [RMS95, SR98]. Bien que cette approche soit attractive, elle présente différents inconvénients qui ont limité sa popularité. Tout d'abord, l'apprentissage de systèmes de régulation génétique via l'estimation de paramètres n'est possible que dans des modèles linéaires, l'intérêt de ces derniers étant que leurs paramètres caractérisent la structure même du réseau de régulation. Pour des équations différentielles non linéaires telles que celles que nous avons vues dans la sous-section 3.2, l'estimation des paramètres constitue un problème épineux qui ne permet de caractériser que la nature des interactions régulatrices dont l'existence doit être connue à l'avance. Dans ce qui suit, nous allons commencer par nous intéresser à l'apprentissage de modèles linéaires. Nous nous attarderons ensuite sur la question épineuse de la taille des jeux de données utilisés pour cet apprentissage. Nous parlerons ensuite de l'apprentissage de modèles non linéaires. Enfin, nous évoquerons brièvement les modèles hybrides qui se situent à mis chemin des modèles différentiels linéaires et les réseaux booléens présentés dans la section 3.3.
L'étude des modèles différentiels est une discipline à part entière qui s'appuie sur de nombreux outils mathématiques et algorithmiques complexes. Dans la mesure où cette classe de modèle est fort éloignée de celle que nous avons utilisée dans nos travaux, nous ne détaillons pas ici les multiples techniques mises en œuvre pour l'estimation de paramètres.

3.2.3.1 Estimation dans les modèles différentiels linéaires

Une manière de modéliser un réseau de régulation au moyen d'équations différentielles linéaires est la suivante :

$$\frac{dx_i}{dt} = \sum_{j=1}^{n} \lambda_{ij} x_j + \mu u + k_i - \gamma_i x_i \tag{3.11}$$

avec $\mathbf{x}$ le vecteur d'état des concentrations des protéines ou d'ARNm. Les paramètres λ_{ij} et μ décrivent les influences respectives des variables d'état et de la variable d'entrée u sur le taux d'expression de X_i. La constante k_i caractérise le niveau basal d'expression de X_i alors γ_i est le taux de dégradation de cette molécule. Dans ce modèle, on remarquera l'introduction d'un terme

$(\mu \cdot u)$ modélisant l'influence de la concentration u d'une molécule extérieure sur le système, ainsi que l'introduction d'un terme $(-\gamma_i x_i)$ modélisant la dégradation de la molécule X_i. Ces deux termes sont optionnels et ne sont pas systématiquement repris dans la littérature.
Dans la mesure où le signe de λ_{ij} spécifie la nature de l'influence régulatrice exercée par le produit du gène j sur l'expression du gène i, les valeurs des paramètres du modèle permettent de caractériser le réseau de régulation génétique sous-jacent. En effet, lorsque $\lambda_{ij} = 0$, on en déduit que j n'exerce pas d'influence régulatrice sur i. Dans le cas contraire, selon que la valeur de ce paramètre est inférieure ou supérieure à 0, on a une répression ou une induction. Le fait que les paramètres de l'équation (3.11) spécifient *à la fois* la structure du réseau de régulation génétique et la façon dont les gènes interagissent doit être souligné. Grâce à cela, il est possible d'apprendre la structure du modèle — et donc d'identifier les interactions entre gènes — en estimant ses paramètres à partir de données expérimentales.

L'une des principales contributions à l'étude de l'estimation des modèles linéaires est [CHC99]. Dans cet article, Chen et collègues ont proposé un modèle de l'expression génétique fondé sur des équations différentielles linéaires ainsi que deux algorithmes permettant de retrouver les fonctions de régulation à partir d'un petit nombre d'observations. La première méthode utilisant les transformées de Fourier pour les systèmes stables suppose que l'expression des gènes est périodique durant le cycle cellulaire. La seconde visant à trouver les solutions de poids minimum des équations linéaires fait l'hypothèse que le nombre de régulateurs pour chaque gène est faible (inférieur à 10) et constant. Cette dernière hypothèse simplificatrice, qui permet de résoudre efficacement le problème d'estimation des paramètres du modèle, est couramment employée dans la littérature. C'est une contrainte raisonnable [2] qui est fréquemment utilisée pour estimer d'autres types de modèles tels que les réseaux Bayésiens que nous verrons dans le chapitre suivant. La particularité des travaux de Chen et collègues porte sur les données exploitées. Les observations sont des échantillons dynamiques des quantités d'ARN et de protéines représentées au sein du modèle. Bien que cette approche soit prometteuse, il faut noter que pour tenir compte de la disponibilité limitée de jeux de données croisés (comprenant à la fois des mesures des concentrations d'ARN et de protéines) les auteurs ont également proposé une version modifiée de leur méthode reposant uniquement sur la prise en compte de l'une ou l'autre de ces espèces moléculaires (ARN ou protéine) au sein du modèle.

3.2.3.2 La taille des jeux de données en question

Dans [DWFS99], on considère que l'expression d'un gène cible dépend directement de celle de ses régulateurs, les étapes intermédiaires impliquant les protéines par exemple n'étant pas modélisées. La façon dont les expressions d'un ensemble de gènes régulateurs influencent l'expression d'un gène régulé est représentée par un modèle linéaire à temps discret. Pour en arriver là, les auteurs font l'hypothèse que l'équation (3.11) peut être approchée par l'équation suivante :

$$\frac{x_i(t+\Delta t) - x_i(t)}{\Delta t} = \sum_j \lambda_{ij} x_j(t) \tag{3.12}$$

Δt est l'intervalle de temps séparant deux observations. On peut réécrire l'équation précédente de la manière suivante :

$$x_i(t+\Delta t) = \sum_j W_{ij} x_j(t) \tag{3.13}$$

[2] Elle se fonde plus particulièrement sur l'idée que la distribution des degrés dans les réseaux de régulation suit une loi de puissance et que, par conséquent, l'essentiel des gènes ont peu d'interactions avec leur voisins.

où $W_{ij} = (\lambda_{ij} \cdot \Delta t + 1)$ si $i = j$ et $W_{ij} = \lambda_{ij} \cdot \Delta t$ sinon.

Le fait de remplacer un système d'équations différentielles linéaires par un système d'équations linéaires simples permet d'utiliser des techniques classiques d'estimation de paramètres telles que la régression multiple. Cette réécriture du modèle implique cependant que l'on suppose que le laps de temps séparant les différentes observations est petit (classiquement proche de 1) et uniforme. Cela est rarement le cas avec les cinétiques d'expression. Le temps séparant deux expériences est grand en comparaison de la vitesse des réactions biochimiques modélisées. Afin de s'affranchir de ce problème, les auteurs proposent de générer des mesures artificielles permettant de « boucher les trous » entre deux mesures trop espacées dans le temps. Ces données synthétiques sont générées par interpolation cubique des données initiales.
Un problème récurrent lorsque l'on souhaite reconstruire un réseau de régulation concerne le nombre de mesures disponibles pour entreprendre l'estimation du modèle. Le nombre de mesures étant très faible face au nombre de paramètres à estimer, il existe de nombreuses solutions (jeux de paramètres) pouvant expliquer de manière satisfaisante les observations expérimentales. Afin de réduire le nombre de paramètres à estimer, van Someren et collègues [vSWR00] ont proposé de regrouper les gènes présentant des profils d'expression similaires au sein de gènes prototypes. Il s'agit ensuite de caractériser les relations existant entre ces gènes prototypes, sachant que ces dernières sont représentées par des modèles linéaires.

3.2.3.3 Estimation dans les modèles différentiels non linéaires

Les modèles différentiels linéaires peuvent apparaître comme une simplification abusive des réseaux biochimiques. Ils reposent sur l'hypothèse implicite que toutes les interactions régulatrices peuvent être traitées comme des événements indépendants. En effet, dans une combinaison linéaire, l'effet d'un régulateur sur sa cible ne dépend que de son état (de sa concentration) et du poids qui lui est assigné. Pourtant, cette hypothèse est contredite par le fait que de nombreux régulateurs transcriptionnels ont une activité différente en fonction de leur(s) partenaire(s) protéique(s). Ceci est particulièrement vrai pour des protéines régulatrices rentrant dans la constitution d'un polymère. L'approximation linéaire étant peu réaliste, il est naturel de se tourner vers les modèles différentiels non linéaires qui permettent de définir des relations fonctionnelles d'une grande complexité.

Il est possible de proposer un grand nombre de fonctions non linéaires pour définir ce type de modèles. Nous présentons un modèle classique pour la modélisation de réactions biochimiques reposant sur l'équation de Michaelis et Menten [QBdB07]. Nous considérons un exemple dans lequel un gène X_i est induit par une protéine X_j et réprimé par une protéine X_k :

$$\frac{dx_i}{dt} = g^+(p_i^n, V_i^{max}, K_{ij}, n) \cdot g^-(p_k^n, V_i^{max}, K_{ik}, n) \tag{3.14}$$

où $g^+(p_i^n, V_i^{max}, K_{ij}, n) = V_i^{max} \cdot \frac{p_j^n}{K_{ij}^n + p_j^n}$ et $g^-(p_k^n, V_i^{max}, K_{ik}, n) = V_i^{max} \cdot \frac{K_{ik}^n}{K_{ik}^n + p_k^n}$ sont les facteurs d'induction et de répression, respectivement. V_i^{max} est le taux de transcription maximum du gène X_i, K_{ij} et K_{ik} sont les concentrations de protéines à partir desquelles X_i atteint la moitié de son taux de transcription maximum. Enfin, n décrit la forme de la sigmoïde. Comme on peut le constater, la nature activatrice ou inhibitrice de l'influence d'un régulateur sur sa cible se traduit dans la forme même de l'équation. En outre, les contributions des différents régulateurs sont étroitement liées, du fait que l'influence régulatrice globale sur X_i s'écrit comme le produit des influences élémentaires. Le fait qu'il s'agisse d'un modèle multiplicatif et non cumulatif, complique singulièrement l'estimation de ses paramètres. Plus simplement, on constate que contrairement aux modèles linéaires où toutes les variables du système sont représentées, seuls les régulateurs

d'un gène apparaissent dans l'équation non linéaire modélisant la variation de sa concentration. Alors que dans les modèles linéaires le poids précédant chaque variable peut être mise à 0 pour signifier que celle-ci n'a aucune influence sur le gène cible, dans les modèles non linéaires, les régulateurs doivent être connus à l'avance.

L'estimation des paramètres de ces équations ne permet pas de caractériser les interactions entre les molécules représentées. En effet, les paramètres des modèles non linéaires ne « codent » pas directement la structure du système de régulation (comme c'est le cas pour les modèles linéaires), celle-ci étant codée dans la formulation même des équations. Par conséquent, la construction du modèle suppose que les interactions régulatrices sont connues au préalable, les paramètres estimés ne permettant que de caractériser la nature (activatrice ou inhibitrice) et la « force » de ces interactions.
D'une manière générale, l'estimation de paramètres dans les systèmes d'équations différentielles non linéaires pose de réelles difficultés. Ce sujet est traité indépendamment des questions biologiques par Ramsay et collègues [RHCC07] qui avancent que la plupart des méthodes actuelles d'estimation des EDO à partir de données bruitées sont lentes et fournissent des résultats peu fiables. Différents travaux se sont également intéressé à cette problématique en biologie. Dans [MMB03] par exemple, différentes méthodes d'optimisation sont comparées pour estimer les paramètres de modèles non linéaires de réseaux biochimiques. Dans [vRS06], la théorie du contrôle est utilisée pour définir un plan d'expérience et identifier les paramètres de modèles non linéaires représentant des mécanismes de transduction du signal et des voies métaboliques.

3.2.3.4 Les modèles hybrides : entre modèles linéaires et modèles logiques

Dans [NST+98], un modèle de réseaux pondérés se situant à mi-chemin entre les équations différentielles linéaires à temps discret et les réseaux booléens (abordés dans la section suivante) est présenté. Chaque gène ne pouvant prendre que deux valeurs, l'état d'un gène X_i à l'instant t dépend de la somme pondérée des états (binaires) de ses régulateurs à l'instant $t-1$. Ces poids sont associés à chacun des arcs du graphe orienté représentant les interactions entre gènes. Selon qu'un poids est positif ou négatif, l'influence correspondante est une activation ou une inhibition. S'il est nul, il n'y a pas d'interaction. Ces poids caractérisent la nature et la force de chacune des influences régulatrices du système. Par ailleurs, un paramètre de seuil est associé à chaque gène afin de décider de son état en réponse à la régulation. Si la somme pondérée des états des gènes régulateurs du gène X_i est supérieure au seuil d'activation de ce dernier alors $x_i = 1$, sinon, $x_i = 0$. Afin de reconstruire le réseau de régulation grâce à ce modèle, les auteurs proposent un algorithme d'ajustement itératif des matrices des poids.
Dans [WWS99], Weaver et collègues mettent en exergue la nécessité de pouvoir attribuer plusieurs états à un gène. Il en résulte un modèle dynamique, discret et linéaire (avec mise à jour synchrone des états des variables du modèle) clairement inspiré par les réseaux de neurones : une fonction sigmoïdale permet d'intégrer les influences régulatrices en entrée (qui reposent sur une combinaison linéaire des états des gènes régulateurs) pour établir l'état du gène régulé en sortie.

3.3 Les modèles logiques

Comme nous l'avons vu à plusieurs reprises, la richesse offerte par les modèles différentiels constitue souvent un handicap pour leur élaboration et pour leur analyse. Nous avons notamment remarqué qu'il était difficilement envisageable de réunir les nombreuses informations, souvent

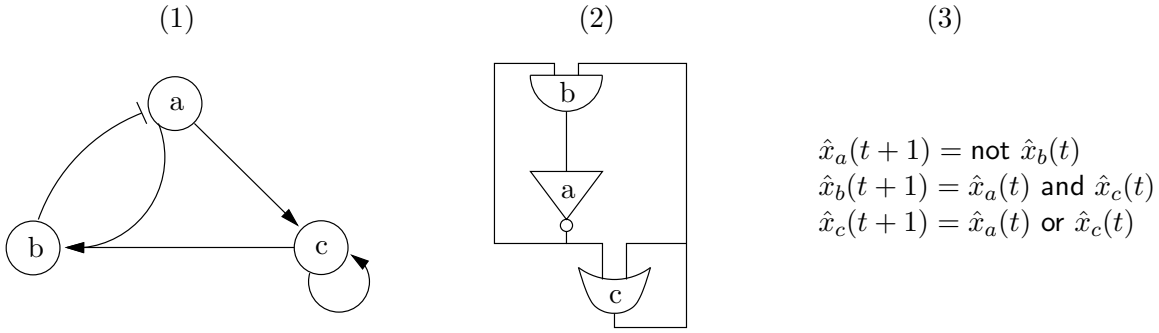


Fig. 3.4 – Un réseau de régulation génétique modélisé par un réseau booléen. (1) Un réseau de régulation génétique représentant les influences régulatrices entre gènes. (2) Le circuit logique correspondant au réseau de régulation génétique. (3) Les fonctions booléennes définissant l'état de chaque gène au temps $t+1$ en fonction de celui de ses régulateurs au temps t.

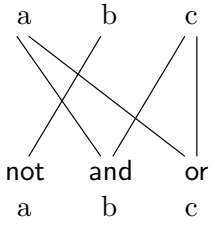
très précises, nécessaires à la représentation des multiples mécanismes moléculaires impliqués dans un réseau de régulation génétique. Afin de simplifier la construction de modèles pertinents ainsi que la caractérisation de leur comportement dynamique, il est possible de se tourner vers des modèles plus simples, plus qualitatifs, tels que les réseaux booléens initialement introduits par Kauffman [Kau69, GK73, Kau74] et les modèles logiques généralisés introduits par Thomas [Tho98, TTK95].

3.3.1 Les réseaux booléens

L'utilisation de fonctions constantes par morceaux dans les équations des taux de réaction permet en effet d'envisager une simplification plus drastique dans la description des éléments des réseaux de régulation génétique et de leurs interactions.
L'approche la plus radicale consiste à décrire l'état de chaque gène X_i par une variable booléenne $\hat{x}_i$ prenant la valeur 0 ou 1 selon que le gène X_i est respectivement inactif ou actif. Lorsque $\hat{x}_i = 1$, on suppose que le produit final du gène est présent, alors qu'il est absent lorsque $\hat{x}_i = 0$. On fait donc l'hypothèse implicite que les différentes étapes de la synthèse des protéines ainsi que les différentes espèces moléculaires qui y participent peuvent être intégrées au sein d'une même variable, sans qu'il soit nécessaire de rendre compte des différents niveaux de régulation possibles. À partir de là, l'influence exercée par un ensemble de gènes régulateurs sur un gène cible peut être spécifiée par une fonction booléenne qui calcule l'état binaire du gène cible en fonction de l'état de ses régulateurs. On définit ainsi un réseau booléen modélisant un réseau de régulation génétique. Un exemple est présenté à la figure 3.4. L'approximation booléenne [SF63] a donné lieu à divers travaux en biologie des systèmes [Kau93].

Soit $\hat{\mathbf{x}} = \{\hat{x}_1, \ldots, \hat{x}_i, \ldots, \hat{x}_n\}$ un vecteur binaire représentant l'état des n éléments d'un réseau de régulation génétique. Chaque $\hat{x}_i, 1 \leq i \leq n$ pouvant prendre la valeur 0 ou 1, le nombre d'états possibles pour le système est 2^n. L'état $\hat{x}_i$ d'un gène au temps $t+1$ correspond à la valeur de la fonction booléenne $\hat{b}_i$ calculée à partir de l'état de k gènes parmi n au temps t. Notons que tous les gènes n'ont pas forcément le même nombre k de gènes régulateurs. La variable $\hat{x}_i$ décrivant l'état du gène cible i est appelée variable de sortie de $\hat{b}_i$. Les k variables décrivant l'état des gènes régulant le gène i sont désignées comme les variables d'entrée de $\hat{b}_i$.

Pour k variables d'entrée, le nombre de fonctions booléennes $\hat{b}_i$ possibles est égal à 2^{2^k}.



	$\hat{x}_a$	$\hat{x}_b$	$\hat{x}_c$
t_0	1	1	1
t_1	0	1	1
t_2	0	0	1
t_3	1	0	1
t_4	1	1	1

Fig. 3.5 – Un diagramme de WIRING. La liste de variables situées dans la partie supérieure du diagramme représente l'état du système au temps t, alors que celle située dans la partie inférieure représente l'état du système au temps $t+1$. Dans la partie inférieure, les fonctions logiques déterminant l'état des variables au temps $t+1$ en fonction de leurs régulateurs au temps t sont rappelées. À droite, la table de transition représente les états successifs du système. On remarque que le système retourne à son état initial au bout de 4 pas de temps.

En effet, on compte 2^k configurations possibles pour le vecteur d'état des variables d'entrées, chacune d'entre elles pouvant donner soit la valeur 1 soit la valeur 0 à la variable de sortie selon la fonction $\hat{b}_i$ choisie. Le nombre de fonctions booléennes envisageables pour décrire la régulation d'un gène X_i croît donc de manière super-exponentielle avec le nombre k de régulateurs de i. Cela complique singulièrement la construction d'un réseau booléen et ce, même à structure fixée (c'est-à-dire la liste des régulateurs de chaque gène étant connue). Pour chaque élément du système, le choix de la fonction booléenne adéquate est particulièrement vaste.

En résumé, la dynamique d'un réseau booléen décrivant un réseau de régulation est donnée par :

$$\hat{x}_i(t+1) \;=\; \hat{b}_i(\hat{\mathbf{x}}(t)),\; 1 \leq i \leq n, \tag{3.15}$$

où $\hat{b}_i$ est une fonction de k variables. La figure 3.4 donne également les équations booléennes décrivant la dynamique du réseau booléen représenté.La structure d'un réseau booléen peut être redéfinie sous la forme d'un diagramme de WIRING dont un exemple est fourni à la figure 3.5 avec une table présentant les transitions du système sur 4 pas de temps. Cette représentation est particulièrement adaptée au calcul des transitions entre états du système.Les fonctions booléennes de tous les éléments du système étant appliquées simultanément à leurs entrées, toutes les variables de sortie sont mises à jour simultanément. Les transitions entre états d'un réseau booléen sont donc synchrones. Ces transitions sont également déterministes car à un état en entrée ne correspond qu'un seul et unique état en sortie.
La détermination des *états attracteurs*, des *états transitoires* ou des *bassins d'attraction* du système permet de caractériser la dynamique d'un réseau booléen. Pour des réseaux de petite taille (quelques gènes), il est possible de calculer les attracteurs, ainsi que les bassins d'attraction à la main, mais pour des réseaux plus vastes (quelques dizaines de gènes) l'utilisation de logiciels spécialisés est nécessaire.

Dans le formalisme des réseaux booléens, des hypothèses simplificatrices assez fortes quant à la structure et à la dynamique des réseaux de régulation génétique sont faites. Notamment, un gène n'admet que deux états, actif ou inactif, tous les niveaux d'expression intermédiaires étant négligés. Par ailleurs, les transitions entre les états des différents gènes se font de manière déterministe et synchrone. De nombreux comportements ne peuvent être prédits du fait de ces

simplifications parfois abusives. Dans les faits, les transitions notamment, ne se font pas de manière simultanée, le temps nécessaire à l'exécution des mécanismes de régulation pouvant varier en fonction des interactions régulatrices concernées. On peut donc s'interroger quant à la pertinence d'une mise à jour synchrone des variables [FNCT06] pour étudier la dynamique des réseaux de régulation biologique.

3.3.2 Les formalismes logiques généralisés

Afin de dépasser certaines limites inhérentes aux réseaux booléens, René Thomas et collègues ont développé une approche logique généralisée fondée sur un formalisme à mi-chemin entre les réseaux booléens et les équations différentielles linéaires par morceaux. Il s'agit d'un formalisme discret, permettant d'une part d'abstraire et de simplifier la description continue des équations différentielles et d'autre part de généraliser les réseaux booléens en permettant à chaque variable d'avoir plus de deux valeurs et aux transitions entre états d'être effectuées de manière asynchrone. Depuis sa conception, cette méthode a connu de nombreuses extensions. Des explications plus détaillées de la version que nous présentons ici peuvent être trouvées dans la littérature [Tho98, TTK95].

Le formalisme de Thomas utilise des variables $\hat{x}_i$ appelées *variables logiques* qui sont des abstractions des variables de concentration x_i employées dans les modèles différentiels. Dans ce cadre, on suppose que les influences régulatrices élémentaires peuvent être modélisées par des fonctions de Hill. Tout comme dans le cas des modèles différentiels linéaires par morceaux, on fait l'hypothèse que ces sigmoïdes peuvent être approcher par des fonctions à seuil. Les valeurs pouvant être prises par une variable $\hat{x}_i$ sont déterminées en fonction de la concentration x_i nécessaire pour franchir le seuil à partir duquel X_i régule un autre élément du système.

Si un élément X_i exerce une influence sur p éléments du système de régulation, on s'attend à ce qu'il ait p seuils à franchir pour rendre ces influences régulatrices effectives.

$$\sigma_i^{(1)} < \sigma_i^{(2)} < \ldots < \sigma_i^{(p)}$$

$\hat{x}_i$ peut prendre les valeurs $\{0, \ldots, p\}$ nécessaires au franchissement de cet ensemble de seuils et est définie comme suit :

$$\begin{aligned} &\hat{x}_i = 0, \text{ si } x_i < \sigma_i^{(1)} \\ &\hat{x}_i = 1, \text{ si } \sigma_i^{(1)} < x_i < \sigma_i^{(2)} \\ &\ldots \\ &\hat{x}_i = p, \text{ si } \sigma_i^{(p)} < x_i \end{aligned} \tag{3.16}$$

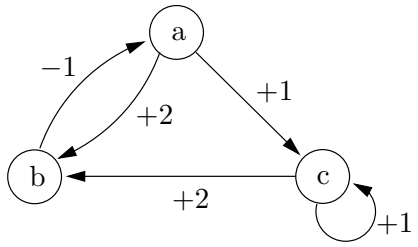
Le vecteur $\hat{\mathbf{x}}$ représente l'état logique du système de régulation.

Les influences régulatrices du système sont décrites au moyen d'équations logiques de la forme :

$$\hat{x}_i^{im}(t) = \hat{b}_i(\hat{\mathbf{x}}(t)),\ 1 \leq i \leq n, \tag{3.17}$$

où $\hat{x}_i^{im}$ est appelé *image* de $\hat{x}_i$. Le graphe orienté étiqueté ainsi que les équations logiques qui l'accompagnent constituent un réseau multivalué.
L'image est la valeur vers laquelle tend $\hat{x}_i$ lorsque l'état logique du système au temps t est $\hat{\mathbf{x}}$. Il est intéressant de la distinguer de la dérivée par rapport au temps de l'équation (3.1) propre aux EDO et de la valeur du successeur de l'équation (3.15) dans les réseaux booléens. Par ailleurs, la fonction logique $\hat{b}_i$ est une généralisation de la fonction booléenne de l'équation (3.15) du fait que les variables logiques peuvent à présent prendre plus de deux valeurs. La fonction logique

(1)



(2)

$$\hat{x}_a^{im} = \hat{b}_a(\hat{x}_b)$$
$$\hat{x}_b^{im} = \hat{b}_b(\hat{x}_a, \hat{x}_c)$$
$$\hat{x}_c^{im} = \hat{b}_c(\hat{x}_a, \hat{x}_c)$$

$\hat{x}_b$	$\hat{b}_a(\hat{x}_b)$
0	2
1	0

	$\hat{b}_b(\hat{x}_a, \hat{x}_c)$	$\hat{x}_c$ = 0	1	2
$\hat{x}_a$	0	0	0	0
	1	0	0	0
	2	0	0	1

	$\hat{b}_c(\hat{x}_a, \hat{x}_c)$	$\hat{x}_c$ = 0	1	2
$\hat{x}_a$	0	0	0	0
	1	0	1	1
	2	0	1	2

(3)

Fig. 3.6 – Un exemple de réseau multivalué. (1) Un réseau de régulation génétique multivalué. Les gènes a, b et c prennent respectivement 3, 2 et 3 valeurs. Par exemple, le gène a régulant deux gènes, il prend les valeurs 0, 1 ou 2 selon qu'il n'en régule aucun, qu'il régule l'un des deux ou qu'il régule les deux. Les arcs représentant les influences régulatrices entre gènes sont signés et étiquetés. Le signe correspond au type de régulation : induction ou répression. L'étiquette représente la valeur à partir de laquelle un gène régule sa cible : a régule c quand il passe de la valeur 0 à 1, il régule b uniquement lorsqu'il prend la valeur 2. (2) Les équations logiques déterminant l'image de chaque variable en fonction de l'état de ses régulateurs. (3) Les tables de paramètres détaillant le comportement des équations logiques.

$\hat{b}_i$ calcule l'image de $\hat{x}_i$ à partir de l'état du système, plus spécifiquement à partir de la valeur des k éléments régulateurs de i. Un exemple de réseau multivalué accompagné des équations logiques correspondantes et des tables de paramètres spécifiant leurs comportements est fourni à la figure 3.6.

Les équations logiques sont à la base de l'analyse de la dynamique du système de régulation, et plus particulièrement de la détermination des états logiques stables [ST92]. Un état logique stable apparaît quand l'état logique du système est identique à son image $\hat{X}_i^{im} = \hat{x}_i$. Comme le nombre d'états logiques est fini, il est possible de rechercher de manière exhaustive les états logiques stables, ainsi que les bassins d'attraction, en étudiant les transitions entre états logiques [Tho91]. Dans ce formalisme, les transitions entre états logiques s'effectuent de manière asynchrone. En effet, le laps de temps nécessaire à l'exécution d'une réaction biochimique ou à la diffusion des molécules (enzymes ou substrats) à travers la cellule étant variable, il semble naturel que les molécules du système aient des délais de mise à jour de leur concentration différents. Par conséquent, il y en a toujours une qui précède les autres et l'état courant du système a alors pour successeur l'état dans lequel seule cette variable à été mise à jour. En supposant que les variables logiques ne changent pas de valeur simultanément, on peut donc atteindre un maximum de n états successeurs à partir d'un état donné. Lorsque l'on ignore l'ordonnancement des délais de mise à jour des différentes variables logiques, on est amené à considérer tous les états successeurs

consistant en la mise à jour d'une seule variable logique à chaque fois. Bien sûr, si un état logique est stable, il n'a pas de successeur car il est identique à son image. Les états logiques ainsi que les transitions entre ces derniers peuvent être organisés et représentés par un graphe (de transitions) d'états qui constitue le support privilégié pour l'analyse de la dynamique du système, la caractérisation de ses états stables et de ses bassins d'attraction. Enfin, cette analyse dynamique permet de définir les conditions nécessaires à la multi-stationnarité d'un système [RC07] qui est souvent associée à certains phénomènes biologique tels que la différentiation. Notons qu'il est également possible d'enrichir cette analyse par la prise en compte explicite des délais de mise à jour des variables logiques occasionnés par la transcription, la traduction et le transport des molécules d'intérêt [ABC$^+$06].

Les réseaux multivalués assurent un bon compromis entre les EDLM et les réseaux booléens. Ils permettent de représenter de manière abstraite et qualitative des systèmes complexes, dont les caractéristiques moléculaires nous sont rarement connues, tout en étant plus expressifs que les réseaux booléens. Surtout, ils proposent une approche asynchrone de la dynamique des systèmes de régulation qui est plus proche de la réalité. En somme, les réseaux multivalués de Thomas offrent une représentation graphique aisément interprétable ainsi que des outils puissants de simulation et de prédiction. Ces avantages ne doivent pas faire oublier qu'au même titre que les réseaux booléens, cette classe de modèles tourne le dos à un certain nombre de propriétés caractéristiques des systèmes biologiques. La plus importante d'entre elles est sans nul doute leur caractère stochastique (que les équations différentielles stochastiques pouvaient modéliser). Le caractère déterministe des modèles logiques constitue l'une de leurs principales faiblesses dans le cadre de l'apprentissage automatique de modèles.

3.3.3 Apprentissage de modèles logiques

La reconstruction de modèles logiques concerne essentiellement les réseaux booléens. En effet, à notre connaissance, il n'existe pas à l'heure actuelle de méthode permettant de caractériser les modèles discrets de Thomas à partir de données expérimentales.

Les réseaux booléens, initialement introduits par Kaufman [Kau69], furent l'un des premiers formalismes pour lequel on ait développé des méthodes d'induction de modèles [AKMM98a, AMK99, AKMM98b, ITK00]. Malgré leur simplicité, ils permettent de rendre compte du comportement dynamique de systèmes régulatoires complexes à partir d'importants jeux de cinétiques d'expression [KLP07, Hua99]. Ils permettent ainsi l'étude des interactions logiques entre gènes en l'absence de connaissances spécifiques [KLP07, SDZ02a]. L'algorithme *REVEAL* de Liang et collègues [LFS98] est un exemple célèbre d'outil d'inférence de réseaux booléens pour la reconstruction de réseaux de régulation. Dans leur principe ces algorithmes utilisent l'information mutuelle (ce concept est formalisé dans la section 3.4.3.3 page 73) entre les états d'entrée et de sortie du système pour identifier les paires de gènes connectés au sein du réseau. Ils identifient ensuite les fonctions booléennes permettant de spécifier la logique de ces interactions à partir des données. Le découverte de ces fonctions booléennes repose sur des approches purement combinatoires qui recherchent les fonctions de transition booléennes cohérentes avec les trajectoires observées. Cette approche n'est applicable qu'à des réseaux de taille et de complexité restreintes. Typiquement, l'algorithme REVEAL permet de traiter un réseau avec $n = 50$ gènes et $k = 3$ régulateurs par gènes. Liang et col. avancent également que leur algorithme est généralisable à des modèles discrets, dont les variables ont un nombre d'états supérieur à deux. Pour mieux cerner les opportunités d'application de ces méthodes aux mesures de la génomique fonctionnelle, Akutsu [AMK99] a produit une analyse du problème d'identification de réseaux booléens à partir de données obtenues par suppression ou sur-expression multiple de gènes, en fonction

du nombre d'expériences et de la complexité de celles-ci. Il confirme qu'il est possible d'identifier des réseaux booléens à partir d'une collection d'états de transitions incomplètes (100 transitions de type entrées ← sorties, pour un réseau de $n = 100000$ sommets) si le degré entrant maximum des réseaux appris est fortement limité ($k \leq 2$).

Malgré leurs nombreux avantages, les modèles booléens présentent cependant un inconvénient majeur : il s'agit de modèles déterministes, peu à même de traiter des données bruitées et de modéliser des systèmes stochastiques. Les réseaux booléens probabilistes permettent de répondre à ce problème. Ces derniers ont initialement été introduits par Shmulevich et collègues [SGH+03, SDKZ02] et sont une extension stochastique des réseaux booléens. Ils ont été utilisés dans le cadre de la biologie des systèmes [SDZ02b, LSYH03] et différentes méthodes d'apprentissage ont également été explorées [SDZ02a, SMF07].

3.4 Les représentations graphiques

Lorsque l'on ne souhaite pas modéliser l'aspect temporel des réseaux de régulation ou que l'on n'est pas en mesure d'apprendre des modèles dynamiques, il est naturel de se tourner vers les modèles statiques. Dans le cas le plus simple (lorsqu'on ne s'intéresse qu'à l'existence des influences régulatrices et non à leurs effets) l'utilisation de graphes orientés ou non orientés est pertinente. Ces derniers sont définis comme suit :

Définition 3.1 (graphe orienté)
Un graphe orienté est un couple (V, E) où V est un ensemble de sommets et E est un ensemble d'arcs. Un arc $e \in E$ est un couple de sommets (i, j) où $i, j \in V$. On appelle i l'origine de l'arc et j son extrémité.

Définition 3.2 (graphe non orienté)
Un graphe non orienté est un couple (V, A) où V est un ensemble de sommets et A est un ensemble d'arêtes. Une arête est une paire $< i, j >$ où $i, j \in V$.

Dans le contexte des réseaux de régulation génétique, chaque sommet du graphe correspond à un élément du système de régulation (gène, ARN, protéine ou métabolite) alors que les arêtes représentent les interactions entre ces éléments. Lorsque ces dernières sont orientées (on parle alors d'arcs) elles permettent de distinguer la molécule régulatrice de la molécule régulée : un arc $X_i \rightarrow X_j$ représente une influence régulatrice exercée par la molécule X_i sur la molécule X_j. Il est possible d'enrichir le graphe d'un réseau de régulation génétique en étiquetant ses sommets ou ses arcs afin d'incorporer des informations supplémentaires sur les éléments du système de régulation ou leurs interactions. Il est notamment courant de définir les arcs comme des triplets (X_i, X_j, s) où s est le signe de l'influence régulatrice exercée par X_i sur X_j : si $s = +$ il s'agit d'une induction (activation), si $s = -$ il s'agit d'une répression (inhibition).

3.4.1 Les graphes dans les bases de connaissances

La plupart des bases de connaissances réunissant les informations actuellement disponibles sur les réseaux de régulation biologique s'appuient sur des représentations graphiques (orientées ou non orientées) richement annotées. C'est notamment le cas de GeneNet [APS+02] qui est conçue pour décrire formellement les composants des réseaux de régulation génétique (les gènes, les protéines et les complexes protéiques) ainsi que les interactions et réactions biochimiques entre ces derniers à travers différents organismes et types cellulaires. On peut également citer KEGG pathway [KAG+08] qui présente la particularité d'intégrer voies métaboliques, voies de

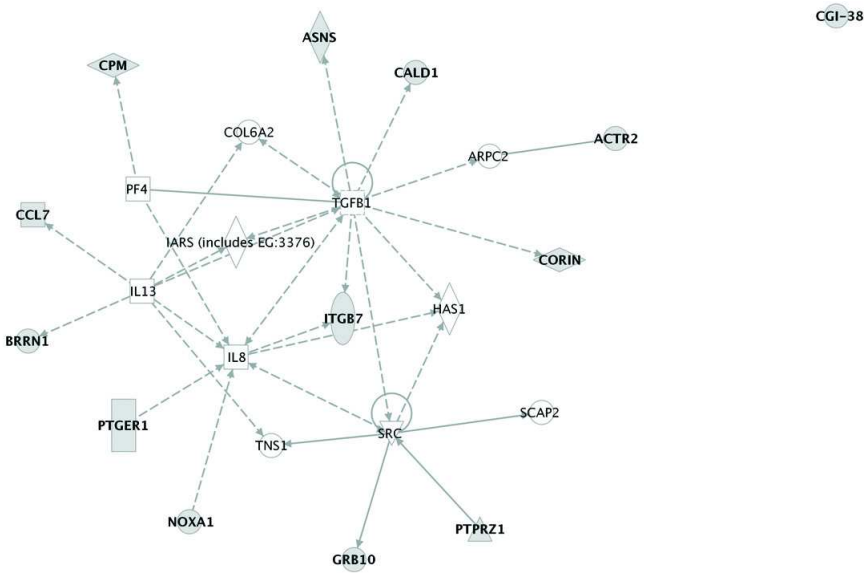


Fig. 3.7 – Exemple de réseau de régulation fourni par le logiciel INGENUITY.

transduction du signal et processus de régulation génétique au sein des mêmes diagrammes (voir la figure 3.4.1). Enfin, on peut citer INGENUITY, une solution commerciale dont le principe est différent des deux solutions libres évoquées précédemment, puisqu'il s'agit de proposer à l'utilisateur des graphes orientés faisant apparaître un arc entre les couples de gènes ou de protéines entre lesquels une interaction est décrite dans la littérature. Ici, le graphe (voir figure 3.7) ne met en évidence que l'existence d'une interaction entre deux entités moléculaires. La nature de cette interaction n'est explicitée que dans la littérature ayant servie de base à la construction de ce graphe.

L'utilisation intensive de graphes dans les bases de connaissances met clairement en évidence le premier attrait de ce type de formalismes : offrir une représentation abstraite et explicite des nombreux mécanismes concourant au fonctionnement des réseaux de régulation génétique. En effet, pour des utilisateurs peu familiers des formalismes mathématiques plus complexes (équations différentielles, réseaux booléens) vus précédemment, les graphes constituent un support aisément interprétable.

3.4.2 Étude des propriétés structurelles des graphes d'interactions

Les graphes ne sont pas de simples outils descriptifs. Il est également possible d'analyser ces derniers au moyen d'outils mathématiques ou algorithmiques issus pour l'essentiel de la théorie des graphes [RC03, CLRS01] ou de l'étude des graphes aléatoires [RPKL07]. Par exemple, la recherche des chemins entre deux gènes du réseau nous renseigne sur l'existence d'une influence régulatrice entre ces derniers et sur l'éventuelle redondance de cette influence. Il est ainsi possible de prévoir si l'activation ou l'inactivation d'un gène est susceptible de perturber le fonctionnement d'un autre gène du réseau. Il s'agit là d'une utilisation directe de la structure des graphes mais d'autres aspects peuvent également être pris en compte.

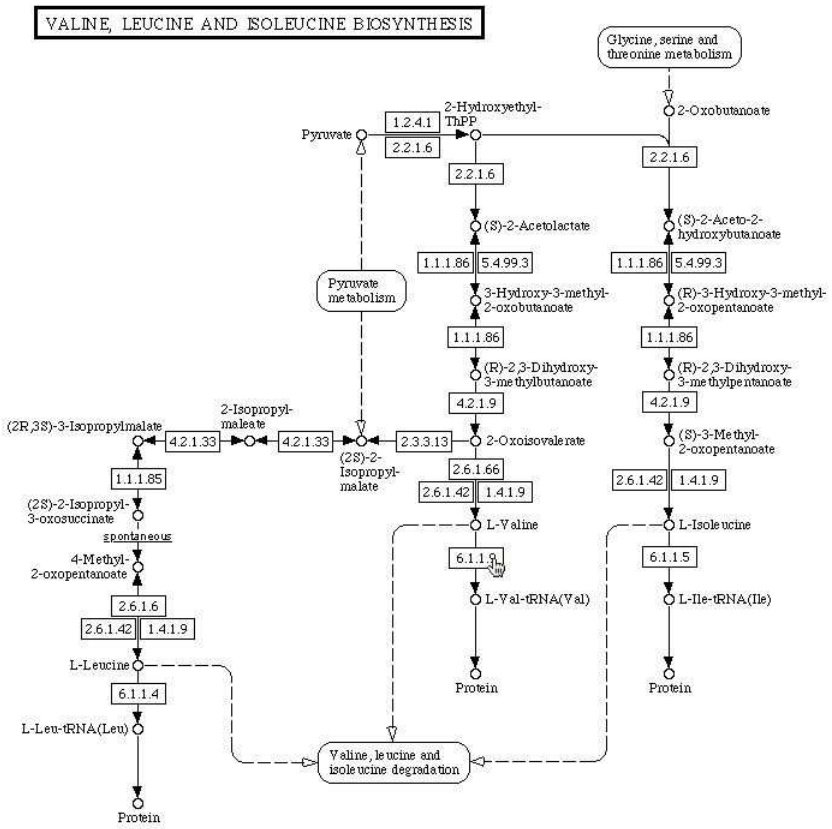


Fig. 3.8 – Exemple de voie métabolique au sein de Kegg. Voie de biosynthèse de trois acides aminés : Valine, Leucine et Isoleucine.

3.4.2.1 Les propriétés globales des graphes d'interactions

En étudiant les propriétés structurelles globales des réseaux de régulation biologique, on a pu montrer qu'ils présentent certaines des caractéristiques structurales rencontrées dans la plupart des réseaux réels tels que les réseaux sociaux. Quelques unes des principales propriétés qui leur sont généralement attribuées sont les suivantes.

Une distribution en *loi de puissance* des degrés des sommets Cela signifie que $P(k)$, la probabilité pour qu'un sommet ait k voisins, diminue de manière exponentielle avec la valeur de k [BA99]. En somme, alors que beaucoup de sommets interagissent avec un petit nombre d'éléments, un petit nombre d'entre eux interagissent avec la plupart des composants du réseau. Ces sommets sont généralement appelés « hubs ». On leur attribue fréquemment un rôle de point de contrôle et d'intégration des informations circulant au sein du réseau de régulation.

Des réseaux de type *petit monde* La plupart des sommets sont proches les uns des autres dans le sens où il suffit de parcourir un petit nombre d'arcs ou d'arêtes pour atteindre la plupart des sommets à partir d'un sommet quelconque.

Des réseaux *sans échelle* La distribution des degrés des sommets ne dépend pas de la taille du réseau en nombre de sommets.

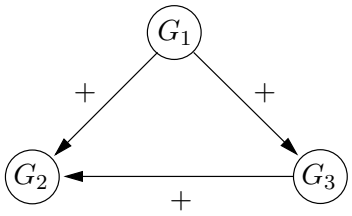


Fig. 3.9 – Un exemple de boucle feed-forward.

3.4.2.2 La recherche de modules fonctionnels

Depuis plusieurs années, de nombreux travaux s'efforcent d'identifier des *modules fonctionnels* au sein des réseaux de régulation génétique. Il s'agit de caractériser de petits sous-graphes dont la topologie apparaît de manière récurrente au sein d'un même réseau ou de réseaux distincts. La répétition d'un motif précis suggère que ce dernier a une fonction biologique particulière, *a fortiori* s'il est peu connecté avec le reste du réseau. C'est par exemple le cas de la boucle *feed-forward* (voir la figure 3.9), jugée importante dans de nombreux mécanismes de régulation, impliquant un filtrage d'événements trop courts dans le temps pour déclencher un effet.

De nombreux travaux ont permis de proposer des méthodes efficaces d'apprentissage statistique de graphe non orientés. Ces derniers sont généralement appelés réseaux d'association [SBA07].

3.4.3 Apprentissage de réseaux d'association

Dans le premier chapitre, nous avons vu que les processus biologiques résultent de l'action concertée de molécules interagissant à différents niveaux. Cette observation a inspiré de nombreux travaux visant à regrouper des profils d'expression similaire au moyen de méthodes de classification [ESBB98, SSZ$^+$98]. L'idée sous-jacente à ces travaux est que des gènes ayant des profils d'expression similaires sont susceptibles d'être co-régulés. Cette idée a été reprise dans le cadre de la reconstruction de systèmes de régulation afin d'inférer des *réseaux d'association*. Il s'agit de modèles non orientés au sein desquels les gènes qui présentent des profils d'expression similaires dans un ensemble de conditions expérimentales sont associés deux à deux au moyen d'une arête. Pour les construire, une mesure de similarité est calculée pour chaque paire de gènes du système. Ces derniers ne sont connectés que si la valeur de leur similarité dépasse un certain seuil. L'aspect fondamental de cette approche réside dans le choix d'une mesure de similarité qui détermine fortement le résultat de la reconstruction. Les choix les plus populaires à cet égard sont des mesures fondées sur la covariance des gènes telle que la corrélation de Pearson [DWFS98, BK99] ou sur la notion d'entropie telle que l'information mutuelle [DWFS98, BTS$^+$00].Pour chaque paire de gènes X et Y, ces mesures visent à établir l'indépendance de X et Y conditionnellement à un sous-ensemble (éventuellement vide) des gènes restants, l'indépendance conditionnelle de X et Y impliquant l'absence d'arêtes entre ces derniers. Avant d'aller plus loin, il est fondamental de préciser la notion d'indépendance conditionnelle qui est centrale dans l'étude des réseaux d'association, mais aussi et surtout pour la compréhension des modèles graphiques orientés que nous présentons dans le chapitre suivant.

Définition 3.3 (Indépendance conditionnelle)
Soient un triplet de variables aléatoires (X, Y, Z) de distribution de probabilité jointe P. On dit que X est conditionnellement indépendant de Y sachant Z (et on le note $X \perp Y \mid Z = z$) si et seulement si pour toute instanciation de (X, Y, Z) notée x, y, z :

$$P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z) \cdot P(Y = y \mid Z = z) \tag{3.18}$$

Par ailleurs, si $X \perp Y \mid Z = z$, alors :

$$P(X = x \mid Y = y, Z = z) = P(X = x \mid Z = z) \tag{3.19}$$

Cette propriété est une généralisation directe de l'indépendance marginale entre deux variables aléatoires X et Y (lorsque $Z = \emptyset$), notée $X \perp Y$:

$$P(X = x, Y = y) = P(X = x) \cdot P(Y = y)$$

A contratio, la dépendance entre deux variables aléatoires X et Y est notée $X \not\perp Y$.

La définition précédente peut être étendue au cas où le conditionnement ne porte pas sur une unique variable Z mais sur un ensemble de variables aléatoires $\mathbf{Z}$. Pour interpréter ce concept, on peut voir une variable aléatoire comme une information énoncée par une personne. $X \perp Y \mid Z = z$ signifie alors : « ayant entendu Z, ce que dit Y m'est inutile pour saisir ce que dit X ». En d'autres termes, « les informations fournies par Z me suffisent pour comprendre X, Y ne m'apportant à ce titre aucun information supplémentaire ». La façon dont les indépendances conditionnelles sont représentées au sein d'un modèle graphique constitue ses *propriétés Markoviennes*.

3.4.3.1 Les réseaux de co-expression

La mesure la plus simple qui puisse être envisagée est la corrélation [DWFS98]. C'est une mesure facile à interpréter qui peut être mesurée avec précision, même lorsque le nombre de gènes est supérieur au nombre d'observations comme c'est fréquemment le cas avec les données d'expression (quelques dizaines de mesures portant chacune sur des milliers de gènes). Dans un modèle Gaussien, deux gènes non corrélés sont statistiquement indépendants. La corrélation constitue une mesure de dépendance linéaire entre variables qui peut être utilisée pour inférer des réseaux de co-expression (également appelés *réseaux de corrélation*).

Si à chaque gène i, on associe une variable X_i dont les valeurs mesurées sont notées $x_i(l)$ pour $l = 1, \ldots, m$, la corrélation de Pearson entre les variables aléatoires X_i et X_j est :

$$R(X_i, X_j) = \frac{\sum_{l=1}^{m}(x_i(l) - \bar{x}_i)(x_j(l) - \bar{x}_j)}{(n-1)\sqrt{v_i v_j}} \tag{3.20}$$

où $\bar{x}_i, v_i$ et $\bar{x}_j, v_j$ sont les moyennes et les variances de $x_i(l)$ et $x_j(l)$ pour les m mesures.
Dans la mesure où de nombreux couples de gènes peuvent présenter un comportement similaire uniquement du fait du hasard, il est crucial d'évaluer la significativité des résultats. Il est possible d'estimer un taux de faux positifs en permutant la matrice des données et en comparant la distribution des similarités obtenues à partir de ces jeux de données artificiels au réseau obtenu à partir des données réelles [Bic05]. Il est également possible de recourir à la validation croisée pour choisir les valeurs optimales des seuils à partir desquelles les corrélations sont jugées significatives.

Les réseaux de corrélation ont été utilisés pour l'annotation des gènes [WKB05], les co-expressions identifiées semblant s'accorder avec les similarités fonctionnelles décrites dans *Gene Ontology* [BRCA00]. La capacité des réseaux de co-expression à capturer les propriétés globales des réseaux métaboliques a également été étudiée [SKFW03b, SKFW03a]. Enfin, notons que Bickel [Bic05] a généralisé ces modèles afin de permettre l'utilisation de données cinétiques.

Le principal inconvénient des réseaux de corrélation est que des comportements d'expression similaires apportent peu d'information quant aux mécanismes biologiques à l'œuvre dans la cellule, le concept de corrélation ne permettant pas de distinguer une interaction directe d'une interaction indirecte. À titre d'exemple considérons un triplet de gènes X_i, X_j, X_k dont les expressions sont fortement corrélées. Ces derniers forment une clique au sein du réseau de co-expression que de nombreux mécanismes régulatoires peuvent expliquer [3] : Si l'on s'attarde sur les variables X_i et X_j, on constate que leur corrélation peut s'expliquer par une régulation directe ou indirecte via X_k, mais aussi par un régulateur commun, ce dernier pouvant être soit une variable du système corrélée avec X_i et X_j, soit une variable cachée. Pour trancher, il est possible de modifier directement le système régulatoire afin d'observer les effets d'une intervention ciblée (telle que l'inhibition de X_i, X_k ou X_j) sur l'expression des gènes restants. On peut alors distinguer ces différents modèles car les effets prédits par chacun d'entre eux peuvent être comparés à ceux obtenus à l'issue de l'expérience. Lorsque l'on ne dispose pas de données d'intervention, il est possible de recourir à des méthodes statistiques afin d'être plus précis dans le choix des arêtes que l'on ajoute au modèle. Le concept sur lequel se fondent ces méthodes est celui de l'indépendance conditionnelle que nous avons vu plus haut. Pour décider si une arête entre X_i et X_j doit appartenir au graphe, les modèles statistiques utilisés posent une question de la forme : « X_i et X_j sont-ils conditionnellement indépendants sachant $\mathbf{Z}$? », avec $\mathbf{Z} \subseteq \mathbf{X}$. Théoriquement, X_i et X_j ne sont connectés que si l'on répond à cette question par la négative pour tout sous-ensemble Z de $\mathbf{X}$. Les modèles que nous présentons par la suite diffèrent essentiellement au niveau de la taille de l'ensemble de conditionnement $\mathbf{Z}$. On notera que les réseaux de co-expression sont un cas particulier de ces modèles statistiques ($\mathbf{Z} = \emptyset$), encodant des dépendances marginales.

3.4.3.2 Réseaux d'association et corrélation partielle

Afin de limiter le nombre de faux positifs apparaissant dans les modèles de co-expression, il est possible d'étendre la méthode précédente par la prise en compte de mesures de corrélations partielles [dlFBHM04]. La corrélation partielle minimum d'ordre 1 entre les variables X_i et X_j est obtenue en conditionnant de manière exhaustive la corrélation de X_i et X_j par rapport à toutes les variables X_k telles que $X_k \in \mathbf{X} \setminus \{X_i, X_j\}$. S'il existe $k \neq i, j$ tel que X_k explique la corrélation entre X_i et X_j, alors la corrélation partielle d'ordre 1 entre les variables X_i et X_j devient 0 et les variables X_i et X_j deviennent indépendantes conditionnellement à X_k. Dans ce cas, au sein d'un graphe non orienté, les sommets i et j ne sont pas adjacents mais séparés par k. L'indépendance de X_i et X_j conditionnellement à X_k est notée $X_i \perp X_j \mid X_k$. Formellement, la corrélation partielle minimum d'ordre 1 entre les variables X_i et X_j est :

$$R_{C_1}(X_i, X_j) = min_{k \neq i,j} \mid R(X_i, X_j \mid X_k) \mid \tag{3.21}$$

[3] Ces interactions sont nécessairement des influences régulatrices positives (induction), les régulations négatives (répressions) aboutissant à des variables anti-corrélées : quand l'expression du régulateur est élevée celle du régulé est faible.

où

$$R(X_i, X_j \mid X_k) = \frac{R(X_i, X_j) - R(X_i, X_k)R(X_j, X_k)}{\sqrt{(1 - R^2(X_i, X_k))(1 - R^2(X_j, X_k))}} \tag{3.22}$$

Si $R_{C1}(X_i, X_j) \simeq 0$, il existe k tel que $X_i \perp X_j \mid X_k$.

Parfois le conditionnement par rapport à une seule variable est insuffisant pour démentir une corrélation suggérant la dépendance de deux variables. On est naturellement amené à considérer des corrélations partielles d'ordre supérieur. La corrélation partielle minimum d'ordre 2 entre les variables X_i et X_j est donnée par la formule :

$$R_{C_2}(X_i, X_j) = min_{k,l \neq i,j} \mid R(X_i, X_j \mid X_k, X_l) \mid \tag{3.23}$$

où

$$R(X_i, X_j \mid X_k, X_l) = \frac{R(X_i, X_j \mid X_k) - R(X_i, X_l \mid X_k)R(X_j, X_l \mid X_k)}{\sqrt{(1 - R^2(X_i, X_l \mid X_k))(1 - R^2(X_j, X_l \mid X_k))}} \tag{3.24}$$

On peut continuer de la sorte afin de conditionner les calculs des corrélations par des sous-ensembles de variables de taille croissante. Ce calcul étant réalisé de manière exhaustive pour les n gènes du système, le coût de l'algorithme calculant la corrélation partielle minimum d'ordre k est de l'ordre de $O(n^k)$. Dans la pratique, ce coût devient prohibitif pour des corrélations partielles d'ordre supérieur à deux.

Par ailleurs, pour calculer des indépendances conditionnelles d'ordre élevé avec précision, il faut que le nombre d'observations soit relativement important en comparaison du nombre de variables. C'est du reste l'un des principaux inconvénients des modèles fondés sur des indépendances conditionnelles complètes (d'ordre n), tels que les modèles graphiques Gaussiens décrits plus bas. Ces conditions expérimentales étant rarement remplies, il est courant de recourir à des tests d'indépendance conditionnelle d'ordre 0 et 1 en s'appuyant sur l'hypothèse selon laquelle les réseaux génétiques sont parcimonieux. Tester des indépendances conditionnelles d'ordre 1 implique de ne considérer qu'un triplet de variables à la fois, ce qui constitue un problème de faible dimension ne nécessitant pas de grandes quantités de données. Il faut toutefois justifier cette approche statistique d'un point de vue biologique. Nous avons déjà évoqué les travaux suggérant que la distribution des degrés des sommets d'un réseau de régulation suit une loi de puissance [BA99]. Cela implique que malgré l'existence de hubs peu nombreux et fortement connectés, la majorité des sommets du graphe d'interaction ont peu d'interactions. Divers travaux ont donc exploité la corrélation ainsi que la corrélation partielle d'ordre 1 [WB06, MK04] et d'ordre 2 [dlFBHM04] pour reconstruire des réseaux de régulation. Dans [dlFBHM04], il a été montré comment choisir un seuil permettant de sélectionner au sein de la matrice des poids R les corrélations justifiant la présence d'une arête dans le graphe. La façon dont il est possible de combiner les effets de R, R_{C_1} et R_{C_2} est également abordée.

L'utilisation des corrélations partielles permet d'éliminer du graphe des faux positifs obtenus suite au calcul des corrélations marginales. La corrélation et la corrélation partielle fournissent respectivement une information d'indépendance et d'indépendance conditionnelle : une faible valeur de ces deux mesures pour une paire X_i, X_j garantit l'absence d'arête entre ces deux sommets. Cependant, une valeur élevée des quantités $R(X_i, X_j)$ et $R_{C_1}(X_i, X_j)$ ne garantit pas l'existence d'un arc entre X_i et X_j, la valeur de $R_{C_2}(X_i, X_j)$ pouvant quant à elle être faible ou nulle. En somme, une valeur élevée pour une corrélation partielle d'ordre k entre deux variables X_i et X_j, ne permet pas de conclure à la dépendance de ces dernières. Une corrélation partielle d'ordre supérieur à k peut parfaitement remettre en cause l'existence d'une arête entre

X_i et X_j. C'est pourquoi, malgré les inconvénients pratiques énoncés précédemment, il a été envisagé de construire des réseaux d'association à partir de corrélations complètes d'ordre $n-2$. Les modèles ainsi générés sont des modèles graphiques gaussiens appartenant à la famille des modèles graphiques non orientés.

3.4.3.3 Réseaux d'association et théorie de l'information

Pour construire un réseau d'association, il est possible de remplacer la corrélation de Pearson par des concepts issus de la théorie de l'information tels que l'information mutuelle [BTS+00, MNB+06]. Étant donné une variable aléatoire discrète X_i prenant ses valeurs dans un ensemble $\mathcal{H}_i$, son entropie [Sha48], est définie de la manière suivante :

$$H(X_i) = -\sum_{x_i in \mathcal{H}_i} P(x_i) \log P(x_i) \tag{3.25}$$

où $P(x_i)$ est la densité/distribution de probabilité $P(x_i) = Pr(X_i = x_i), x_i \in \mathcal{H}_i$. L'entropie jointe d'une paire de variables X_i et X_j prenant respectivement leurs valeurs dans les ensembles $\mathcal{H}_i$ et $\mathcal{H}_j$ est égale à :

$$H(X_i, X_j) = -\sum_{x_i \in \mathcal{H}_i, x_j \in \mathcal{H}_j} P(x_i, x_j) \log P(x_i, x_j) \tag{3.26}$$

L'entropie conditionnelle de X_i sachant X_j est définie par $H(X_i \mid X_j) = H(X_i, X_j) - H(X_j)$. L'information mutuelle de X_i et X_j est $I(X_i; X_j) = H(X_i) - H(X_i \mid X_j)$. Elle peut s'exprimer selon la formule

$$I(X_i; X_j) = \sum_{x_i \in \mathcal{H}_i, x_j \in \mathcal{H}_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)} \geq 0 \tag{3.27}$$

Lorsque la distribution de probabilité jointe se factorise en marginale d'ordre 1, c'est-à-dire lorsque les variables sont indépendantes, l'information mutuelle disparaît :

$$P(x_i, x_j) = P(x_i)P(x_j) \Longrightarrow I(X_i; X_j) = 0 \tag{3.28}$$

L'information mutuelle permet donc, au même titre que la corrélation, de rendre compte de la dépendance des variables X_i et X_j, elle est nulle quand ces deux variables sont indépendantes. La différence majeure est que contrairement à la corrélation qui est une mesure de *dépendance linéaire*, l'information mutuelle permet également de mesurer les *dépendances non linéaires* : par exemple, dans le cas où j régule i, elle est non nulle si $X_i = X_j^2$, alors que la corrélation ne rend compte que des dépendances du type $X_i = aX_j + b$ avec $a, b \in \mathbb{R}$. C'est un inconvénient majeur des mesures de similarité à base de corrélation, dans la mesure où il est peu probable que les interactions moléculaires (directes ou indirectes) puissent être systématiquement décrites au moyen de fonctions linéaires.

Lorsque l'information mutuelle est associée à la notion d'indépendance conditionnelle, elle permet de distinguer les associations directes ou indirectes au même titre que la corrélation partielle. L'information mutuelle conditionnée par une tierce variable X_k est :

$$I(X_i; X_j \mid X_k) = H(X_i \mid X_k) - H(X_i \mid X_j, X_k) \tag{3.29}$$

ou de manière équivalente :

$$I(X_i; X_j \mid X_k) = H(X_i, X_k) + H(X_j, X_k) - H(X_k) - H(X_i, X_j, X_k) \tag{3.30}$$

Toutes les paires de sommets peuvent être conditionnées exhaustivement par chacun des $n-2$ sommets restants. Le minimum de ces informations mutuelles conditionnelles peut être utilisé comme mesure de dépendance conditionnelle :

$$I_C(X_i; X_j) = min_{k \neq i,j} I(X_i; X_j \mid X_k) \tag{3.31}$$

Lorsqu'il existe un X_k expliquant la totalité de l'information mutuelle existant entre X_i et X_j, alors le triplet X_i, X_j, X_k a la propriété Markovienne :

$$I(X_i; X_j \mid X_k) = 0 \Longleftrightarrow X_i \perp X_j \mid X_k \tag{3.32}$$

Cela implique que $I_C(X_i; X_j) = 0$, sinon $I_C(X_i; X_j) > 0$.
Tout comme pour la corrélation et la corrélation partielle, les conditions 3.28 et 3.32 peuvent être utilisées pour construire le graphe du réseau d'association des gènes du système. I et I_C peuvent être utilisées de concert, éventuellement avec un seuil (déterminé via une méthode de bootstrap par exemple). Bien qu'il soit possible d'étendre la définition de l'information mutuelle conditionnelle pour prendre en compte un nombre croissant de variables de conditionnement, cela devient infaisable sur le plan pratique dès lors que n est de l'ordre du millier.

Margolin et collègues [MNB+06] ont proposé une variante algorithmique de construction de réseau d'interaction qui utilise l'information mutuelle sans recourir explicitement à un calcul conditionnel. Partant de l'hypothèse que le réseau peut être construit en étudiant les interactions des gènes pris deux à deux, il procède en deux temps. D'abord, l'information mutuelle $I(X_i; X_j)$ est estimée puis sa nullité est testée et lorqu'elle est rejetée, un arc entre X_i et X_j est retenu. Le graphe obtenu est ensuite élagué en utilisant la notion de *Data Processing Inequality* (ou DPI) et consiste à éliminer l'arc correspondant au plus petit élément du triplet $I(X_i; X_j)$, $I(X_i; X_k)$ et $I(X_j; X_k)$ pour tous les triplets possibles $i \neq j \neq k$. Ce dernier traitement permet d'éliminer les arcs indésirables issus de la première étapes et qui traduisaient des interactions indirectes. Cet algorithme, baptisé ARACNe qui a été appliqué à des profils d'expression de lymphocytes B humain [BMS+05] permet de s'affranchir de l'hypothèse gaussienne propre aux approches fondées sur des corrélations partielles.

3.4.4 Les modèles graphiques Gaussiens, un exemple de modèles graphiques non orientés

Les modèles graphiques non orientés (également appelés réseaux Markoviens ou champs aléatoires de Markov) sont des modèles fondés sur des indépendances conditionnelles d'ordre $n-2$. La question à laquelle ils entendent répondre est : « la corrélation observée entre deux gènes peut-elle être expliquée par tous les autres gènes du modèle ? ». Cela revient à dire que deux gènes X_i et X_j dont les niveaux d'expression sont corrélés ne peuvent pas être connectés par une arête si $X_i \perp X_j \mid \mathbf{X} \setminus \{X_i, X_j\}$.
Ces modèles, *a priori* lourds à manipuler du fait de la taille de l'ensemble de conditionnement, deviennent particulièrement simples à traiter dans un cadre Gaussien. Faisons l'hypothèse que les observations sont issues d'une distribution normale multivariée, $N(\mu, \Sigma)$ avec μ les vecteurs des moyennes et Σ la matrice $n \times n$ de covariance des variables du système. Si la matrice Σ est inversible, on peut calculer la corrélation partielle entre X_i et X_j conditionnellement aux $n-2$ gènes restants de manière explicite. Soit $\Omega = \Sigma^{-1}$ la matrice de concentration de la distribution (également appelée matrice de précision), d'éléments $\Omega = (\omega_{ij})$. Alors on appelle coefficient de corrélation partielle entre X_i et X_j la valeur

$$R_{C_{all}}(X_i, X_j) = -\frac{\omega_{i,j}}{\sqrt{\omega_{i,i}\omega_{j,j}}} \tag{3.33}$$

telle que pour $X_i, X_j \in \mathbf{X}$ avec $X_i \neq X_j$:

$$\omega_{ij} = 0 \Longleftrightarrow X_i \perp X_j \mid \mathbf{X} \setminus (X_i, X_j) \tag{3.34}$$

Cette relation est utilisée pour définir les modèles graphiques Gaussiens. Un modèle graphique Gaussien est un graphe non orienté ayant pour sommets les éléments de $\mathbf{X}$. Une arête entre deux variables de ce modèle est définie par l'existence d'un coefficient de corrélation partielle non nul pour ces variables. Les modèles graphiques Gaussiens (GGM) permettent d'éliminer les corrélations marginales élevées pouvant être expliquées par d'autres gènes. Les réseaux de corrélation et les GGM apportent des informations complémentaires. Qu'ils interagissent de manière directe ou indirecte, la plupart des gènes sont susceptibles d'être corrélés. Les coefficients de corrélation sont donc des critères de dépendance médiocres, mais aussi, lorsqu'ils sont nuls, des critères d'indépendance (linéaire) puissants. Inversement, les coefficients de corrélation partielle ont tendance à s'annuler. Ils constituent donc un bon critère de dépendance, mais un faible critère d'indépendance [SS05].

La mise en œuvre des GGM suppose la construction de la matrice de précision et un cadre statistique pour tester la nullité des corrélations partielles. Compte tenu du faible nombre d'observations m par rapport au nombre de gènes n, on se trouve devant les classiques et néanmoins difficiles problèmes de mauvais conditionnement de la matrice Ω et de tests multiples. Afin de contourner ces écueils, l'utilisation des modèles graphiques Gaussiens a d'abord été restreinte à l'évaluation des relations au sein de petits ensemble de gènes [WK00, WMH03] ou entre des clusters de gènes [TH02]. Cette dernière approche n'est pas satisfaisante dans la mesure où il est très difficile d'interpréter des dépendances conditionnelles entre clusters. Surtout, on perd toute l'information relative aux associations entre gènes qui est pourtant l'une des principales motivations pour l'inférence de modèles de régulation. Différents travaux ont donc vu le jour afin d'améliorer les estimateurs des corrélations partielles et d'inférer des modèles graphiques Gaussiens dans un cadre plus réaliste ($m \ll n$) [KW00, SS05]. La plupart d'entre eux sont fondés sur l'idée (déjà développée ci-dessus) que les données d'expression, bien que de grande dimension, sont parcimonieuses (un petit nombre de gènes régulent un gène d'intérêt).
À titre d'exemple, Schäfer et Strimmer ont proposé une procédure d'inférence de modèles graphiques Gaussiens adaptée à de petits échantillons [SS05], qui repose sur trois points. D'abord, un estimateur régularisé de la matrice de variance covariance (dans le cas $n < m$) qui permet le calcul des coefficients de corrélation partielle a été étudié. Un test d'inclusion des arêtes du modèle, qui estime les degrés de liberté de la distribution nulle à partir des données, est proposé. Celui-ci s'inspire des approches Bayésiennes de détection des gènes différentiellement exprimés à partir des données de puces à ADN [ETST01]. Enfin une stratégie de tests multiples utilisant la méthode du taux de faux positifs est employée. Au sein de notre équipe, Tenenhaus [TGGF08] a proposé, dans le cadre des GGM, un estimateur régularisé permettant de calculer les corrélations partielles en utilisant la régression PLS (Partial Least Square).

Chapitre 4

Apprentissage automatique de modèles graphiques orientés

La plupart des modèles que nous avons décrits dans le chapitre précédent ont été développés pour étudier la dynamique des réseaux de régulation. Depuis plusieurs années, les modèles graphiques probabilistes se sont imposés comme le formalisme privilégié pour la modélisation et l'apprentissage des réseaux de régulation génétique. Cette famille de modèles permet de modéliser la nature stochastique des processus de régulation. Elle bénéficie surtout du cadre de l'apprentissage statistique et permet d'exploiter efficacement les données bruitées des profils d'expression. Il s'agit d'un formalisme adapté à l'apprentissage, favorisant l'extraction de la structure de graphes représentant les interactions au sein des réseaux de régulation. Les modèles graphiques permettent de représenter graphiquement une loi jointe, le plus souvent multivariée, et de factoriser celle-ci simplement, en se basant sur la structure de sa représentation graphique. Dans le cas qui nous intéresse, ils modélisent la loi jointe des niveaux d'expression des gènes appartenant aux réseaux de régulation. Le nombre élevé de gènes représentés aboutit à des distributions de probabilité de dimensions élevées qui sont généralement inutilisables en l'état. La factorisation de ces lois jointes en marginales d'ordre inférieur, caractérisant des sous-ensembles de variables indépendantes, est donc nécessaire.

Les modèles graphiques Gaussiens que nous avons vus dans le chapitre précédent sont des modèles graphiques non orientés qui représentent les corrélations partielles entre les variables du système. Malgré leurs nombreux avantages, ces modèles restent assez limités dans la mesure où ils ne représentent que les relations entre les gènes, sans en préciser la nature. Nous nous intéressons donc aux modèles graphiques orientés, et plus particulièrement aux réseaux Bayésiens qui constituent une famille de modèles plus riches. Dans ce qui suit, nous allons présenter ces modèles avant d'offrir un tour d'horizon des diverses approches permettant de les apprendre. L'apprentissage de réseaux Bayésiens étant un domaine de recherche très actif, le nombre de méthodes d'apprentissage actuellement disponibles — tant dans un cadre générique que bio-informatique — est très important. Nous nous concentrerons donc sur la présentation des méthodes qui nous semblent les plus emblématiques dans le cadre que nous sommes fixé : l'exploitation de données de profils d'expression statiques, discrètes et complètes.

4.1 Les réseaux Bayésiens

Jusqu'à maintenant, nous avons vu des méthodes permettant de construire des graphes à partir d'indépendances marginales ($X_i \perp X_j$), d'indépendances conditionnelles d'ordre 1 ($X_i \perp$

$X_j \mid X_k, \forall k \in \mathbf{X} \setminus \{X_i, X_j\}$) ou 2, ainsi que d'indépendances conditionnelles complètes ($X_i \perp X_j \mid \mathbf{X} \setminus \{X_i, X_j\}$). Afin de disposer d'un modèle le plus précis et complet possible, la suite logique est de chercher à déterminer les indépendances conditionnelles à tous les ordres. Dans le graphe que l'on entend générer, deux variables X_i et X_j sont connectées si aucun sous-ensemble des variables restantes ne peut expliquer leur relation (typiquement, leur corrélation) :

$$X_i \perp X_j \mid \mathbf{X_C}, \forall \mathbf{X_C} \subseteq \mathbf{X} \setminus \{X_i, X_j\} \tag{4.1}$$

où $\mathbf{X_C}$ est un ensemble de conditionnement contenant entre 1 et $n-2$ éléments choisis parmi les $n-2$ variables restantes. Cela implique donc de tester exhaustivement les indépendances conditionnelles d'ordre 0 (indépendance marginale) à $n-2$ (indépendance conditionnelle complète). En procédant de la sorte, on s'attend à éliminer les faux positifs et à obtenir un graphe ayant un nombre d'arcs plus restreint qu'avec des tests d'indépendance conditionnelle plus limités. Nous verrons dans la section 4.3.2 page 87 que déterminer les (in)dépendances conditionnelles à tous les ordres permet d'orienter un certain nombre d'arêtes du graphe [SGS+00]. Il en résulte un modèle graphique orienté appelé *réseau Bayésien.*

Les réseaux Bayésiens, initialement introduits par Pearl [Pea88], sont des modèles graphiques orientés qui modélisent une distribution de probabilité jointe sur un vecteur aléatoire $\mathbf{X} = \{X_1, \ldots, X_n\}$. Le nombre de paramètres requis pour définir une loi jointe augmente rapidement avec le nombre de variables aléatoires traitées. Il est donc fondamental d'utiliser les indépendances conditionnelles entre ces variables afin de représenter le modèle de manière compacte et de le rendre plus exploitable.
Formellement, un réseau Bayésien (statique) $\mathcal{B}$ est défini par un couple (S, Θ) dont les composants correspondent respectivement à la structure et à l'ensemble des paramètres du modèle. La structure du réseau S permet de décomposer la loi jointe en marginales d'ordre inférieur et d'expliciter les relations existant entre les variables. Les paramètres Θ caractérisent la nature de ces relations.

Définition 4.1 (Réseau Bayésien)
Un réseau Bayésien offre une représentation graphique de la structure de dépendance entre les composants d'un vecteur aléatoire $\mathbf{X}$. *La structure du modèle repose sur un* graphe orienté sans cycle *(ou DAG pour directed acyclic graph)* $S = (\mathbf{X}, A)$ *où* $\mathbf{X} = \{X_1, \ldots, X_n\}$ *est l'ensemble des sommets de* S *et* $A \subseteq \mathbf{X} \times \mathbf{X}$ *l'ensemble des arcs du graphe orienté tels que ce dernier ne présente aucun cycle orienté (et aucune boucle).*
À chaque sommet de S *est associée une variable* X_i *du vecteur aléatoire* $\mathbf{X}$ [1]. *Les arcs* A *encodent les (in)dépendances conditionnelles entre ces variables. La topologie du graphe permet de définir pour chacun de ses sommets* X_i *un ensemble de parents* $Pa_i = \{X_j \,/\, X_j \in \mathbf{X} \setminus \{X_i\} \text{ et } (X_j, X_i) \in A\}$.
Cette observation nous permet d'énoncer une propriété fondamentale des réseaux Bayésiens : « toute variable est indépendante de ses non descendants conditionnellement à ses parents au sein du DAG ». Cela signifie que connaissant la valeur prise par les parents d'un sommet X_i *de* S, *ce dernier est indépendant de tous ses ancêtres au sein de* S. *Cette propriété permet de factoriser la loi jointe* $P(X_1, X_2, ..., X_n)$ *de la manière suivante :*

$$P(X) = P_S(X) = \prod_{i=1}^{n} P_S(X_i \mid Pa_i, \theta_i) \tag{4.2}$$

[1] Par commodité, nous nous permettrons par la suite d'assimiler directement les variables aléatoires du système aux sommets correspondants.

où $P_S(X_i \mid Pa_i, \theta_i)$ est la distribution de probabilité conditionnelle (CPD) de X_i sachant ses parents Pa_i dans S, ou sa distribution marginale lorsque $Pa_i = \emptyset$. Les distributions de probabilité conditionnelle constituent le deuxième élément permettant de définir un réseau Bayésien et de représenter la loi jointe $P(\mathbf{X})$. Dans la mesure où le type de CPD utilisée pour construire un réseau Bayésien est fixé au préalable par l'utilisateur, le modèle est donc défini par les paramètres $\Theta = \{\theta_1, \ldots, \theta_n\}$ de ces CPD, où θ_i est l'ensemble des paramètres décrivant la distribution locale de la variable X_i.

La factorisation de la loi jointe est l'une des propriétés clefs des réseaux Bayésiens. Elle permet de segmenter l'ensemble des variables d'intérêt en un ensemble de familles [2] qui peuvent être traitées indépendamment. Cette modularité est un atout précieux pour gérer la complexité des systèmes étudiés et réaliser les calculs nécessaires à l'utilisation et à la reconstruction de cette classe de modèles. Un exemple de réseau Bayésien est représenté à la figure 4.1.

Dans le cadre de leur application aux réseaux de régulation génétique, les composantes d'un réseau Bayésien peuvent être caractérisées de la manière suivante :

1. les variables aléatoires $\{X_1, \ldots, X_n\}$ correspondent aux niveaux d'expression d'une collection de gènes $\{1, \ldots, n\}$ appartenant au système de régulation modélisé ;
2. la loi jointe $P(\mathbf{X})$ permet de capturer le comportement global du système de régulation dans un cadre probabiliste ;
3. le DAG S représente la structure du réseau de régulation ;
4. les distributions conditionnelles $P_S(X_i \mid Pa_i, \theta_i)$ sont représentées par un modèle d'interaction local dont la nature dépend du type de relation que l'on souhaite établir entre les « régulateurs » Pa_i et leur cible X_i ;
5. les paramètres Θ permettent de caractériser la nature de ces relations en décrivant la façon dont un ensemble de variables régulatrices influe sur l'expression d'une variable cible.

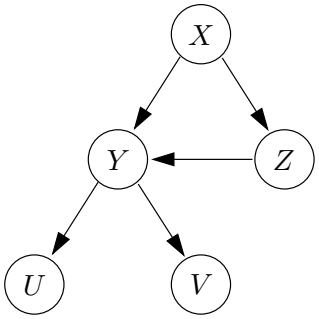
Cette définition basique des réseaux Bayésiens soulève un certain nombre de questions qui sont abordées par la suite. Comment définit-on les distributions de probabilité conditionnelle associées aux sommets du graphe et caractérisant la façon dont l'état d'un gène cible est influencé par l'état de ses régulateurs ? Comment déterminer les indépendances conditionnelles entre ces variables en fonction de la topologie du DAG ? Comment inférer les paramètres et surtout la structure d'un réseau Bayésien à partir de données d'expression ?

4.1.1 Les distributions de probabilité locales d'un réseau Bayésien

Les modèles de réseaux Bayésiens diffèrent par le type de distribution de probabilité conditionnelle, de la forme $P(X_i \mid Pa_i, \theta_i)$, que l'on associe à chaque variable X_i du réseau. Ces dernières dépendent en premier lieu de la nature des variables que l'on considère, discrètes ou continues. Ce choix est lui-même fortement influencé par le type de données que l'on souhaite modéliser. En pratique, on considère deux types de distributions : les distributions multinomiales pour les variables discrètes et les distributions Gaussiennes pour les variables continues.

[2] Une famille est un ensemble de sommets composé d'un sommet et de ses parents au sein du graphe. Formellement, la famille du nœud X_i notée Fa_i est égale à $Pa_i \cup \{X_i\}$

(1)



(2)

$$P(V,U,Y,Z,X) = P(V|U,Y,Z,X) \cdot P(U|Y,Z,X) \cdot P(Y|Z,X) \cdot P(Z|X) \cdot P(X)$$

$$P_S(V,U,Y,Z,X) = P(V|Y) \cdot P(U|Y) \cdot P(Y|Z,X) \cdot P(Z|X) \cdot P(X)$$

(3)

$P(X)$

X	
0	1
0,2	0,8

$P(Z|X)$

		Z	
		0	1
X	0	0,3	0,7
	1	0,5	0,5

$P(Y|Z,X)$

		Y	
Z	X	0	1
0	0	0,6	0,4
0	1	0,8	0,2
1	0	0,3	0,7
1	1	0,5	0,5

Fig. 4.1 – Exemple de réseau Bayésien. (1) La structure d'un réseau Bayésien représentée par un graphe orienté sans cycle. (2) La factorisation de la loi jointe $P_S(V,U,Y,Z,X)$ déduite de la topologie de ce graphe est comparée à une factorisation générique de $P(V,U,Y,Z,X)$, obtenue sans connaissance des indépendances conditionnelles entre les variables. (3) Les distributions de probabilité de 3 variables conditionnellement à leurs parents au sein du graphe sont représentées sous la forme de tables de probabilités conditionnelles. Ces distributions locales ne sont valables que pour des variables discrètes.

Variables discrètes : les tables de probabilité conditionnelle Les variables discrètes suivent une distribution multinomiale paramétrée par des vecteurs de probabilités. On compte un vecteur de probabilités pour chaque configuration parentale, c'est-à-dire pour chaque instanciation de Pa_i. Ces différents vecteurs de probabilités forment une table de probabilité conditionnelle. La probabilité pour qu'une variable prenne chacun de ses états possibles sachant l'état de ses parents peut être calculée à partir des fréquences observées au sein de la base d'apprentissage [3]. Il est courant d'utiliser un *a priori* sur les paramètres afin d'éviter qu'une configuration non observée dans les données ne se voit attribuer une probabilité nulle.

Variables continues : Les distributions Gaussiennes linéaires Les variables continues suivent une distribution Gaussienne $N(\mu, \sigma^2)$ dont la moyenne μ est une combinaison linéaire des états des variables parents. Le plus souvent, la variance σ^2 est constante. Elle peut par exemple être estimée par maximum de vraisemblance, ou bien être fixée par le modélisateur. À titre d'exemple, considérant deux variables continues X_i et X_j, le modèle de régression de la CPD de X_i est de la forme $P(X_i \mid X_j) \sim N(a \cdot X_j + b, \sigma^2)$ où la moyenne μ dépend de la valeur prise par X_j et des constantes a et b déterminée par la régression de X_i sur X_j. Dans ce cas $\theta_i = \{a, b, \sigma^2\}$.

Faire cohabiter des variables discrètes et continues Dans une certaine mesure, il est également possible de combiner des variables discrètes et continues dans un même modèle. Des sommets continus peuvent admettre des parents discrets. Ils suivent alors une distribution Gaussienne dont les paramètres sont fonctions de l'état de leur parents. Comme pour les tables de probabilité conditionnelle des systèmes purement discrets, on a alors un vecteur de paramètres par configuration parentale. Par exemple, pour un sommet continu X_i avec un parent discret X_j ayant $k_j = 3$ états (x_j^1, x_j^2, x_j^3), on a une distribution de probabilité conditionnelle reposant sur k_j distributions Gaussiennes : $N(\mu_1, \sigma_1^2)$, $N(\mu_2, \sigma_2^2)$ et $N(\mu_3, \sigma_3^2)$. Une moyenne μ_l et un écart type σ_l^2 sont les paramètres de la distribution Gaussienne qui modélise $P(X_i \mid x_j^l)$. $\theta_i = \{\mu_1, \sigma_1^2, \mu_2, \sigma_2^2, \mu_3, \sigma_3^2\}$ est l'ensemble des paramètres qui encodent la CPD de X_i grâce aux trois Gaussiennes. Pour estimer θ_i à partir des données il suffit, pour une configuration parentale spécifique, de calculer la moyenne empirique et l'écart type de X_i. Notons qu'a contrario, un sommet discret suivant une distribution multinomiale ne peut admettre de parents continus.

Quelques modèles de distributions locales alternatifs Les modèles présentés ci-dessus sont les plus courants. Il existe cependant un certain nombre de modèles alternatifs assez intéressants en biologie des systèmes.
Segal et collègues [SSR+03] ont par exemple utilisé des arbres de régression. Chaque CPD est représentée par un arbre binaire, dont les nœuds internes correspondent aux parents de la variable d'intérêt et dont les feuilles sont associées avec des distributions Gaussiennes univariées. L'intérêt des arbres de régression est qu'ils permettent de capturer de manière explicite la structure locale des données, alors que le DAG G décrit la structure globale [FGA99].
On peut également utiliser des modèles continus non linéaires [IKG+02] qui permettent de capturer des interactions plus complexes que celles décrites par les distributions Gaussiennes linéaires évoquées plus haut. Nachman et col. [NRF04] ont notamment proposé de modéliser l'influence d'un ensemble de régulateurs sur l'expression d'un gène cible par des équations dynamiques

[3] Ce terme est employé pour désigner les données utilisées pour apprendre un modèle.

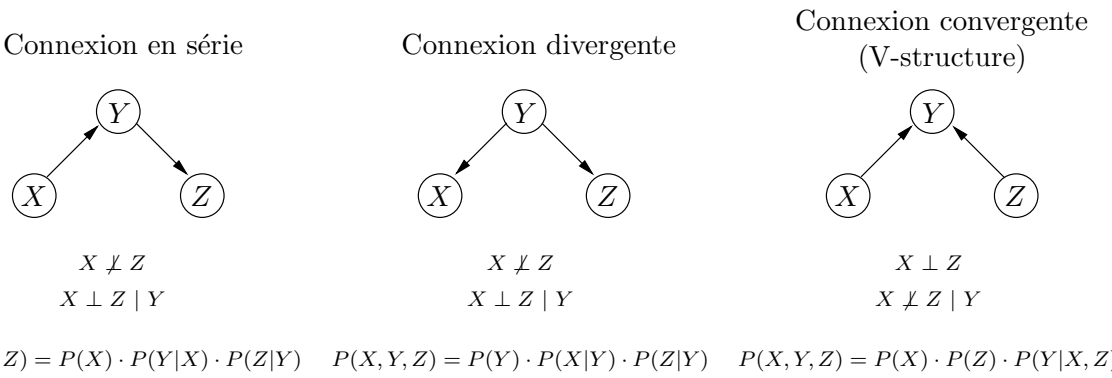


Fig. 4.2 – Illustration des (in)dépendances conditionnelles dans un réseau Bayésien. Dans le cas de la connexion en série et de la connexion divergente, X et Z sont a priori dépendantes ($X \not\perp Z$). Cependant, X et Z sont indépendantes conditionnellement à Y ($X \perp Z \mid Y$). Dans le cas de la connexion convergente (V-structure), X et Z sont indépendantes ($X \perp Z$), cependant Z dépend de X conditionnellement à Y ($X \not\perp Z \mid Y$). Dans les trois cas de figure, X et Y ainsi que Y et Z sont mutuellement indépendantes. Une factorisation résultant des (in)dépendances conditionnelles propre à chaque situation est présenté au bas de la figure.

non linéaires de Michaelis et Mentens. Celles-ci permettent une représentation quantitative et réaliste, sur le plan biochimique, de la régulation génétique. L'inconvénient majeur des modèles non linéaires est que contrairement aux modèles précédents, leur paramètres sont beaucoup plus difficiles à estimer à partir des données.

4.1.2 Les indépendances conditionnelles dans un réseau Bayésien

Comme nous l'avons dit précédemment, la structure d'un réseau Bayésien encode les indépendances conditionnelles entre les variables étudiées. La façon dont cela fonctionne peut être explicitée à travers trois cas de figure présentés ci-dessous. Nous considérons trois modèles comprenant trois variables $\{X, Y, Z\}$, dont la structure et le paramétrage sont supposés connus. Pour simplifier notre propos, nous allons l'illustrer en nous plaçant dans le cas où ces modèles représentent des réseaux de régulation. À la figure 4.2, nous décrivons la situation de ces variables dans chaque cas, au moyen d'un diagramme de réseau Bayésien et de l'équation de la distribution de probabilité jointe qui en découle.

Connexions en série (chaînes) Si la valeur de Y est inconnue, alors le fait de connaître la valeur de X influe sur la valeur attendue de Z. Les variables X et Z sont donc dépendantes. Par contre, si la valeur de Y est fixée, alors le valeur de Z ne dépend que de celle de Y. X et Z sont donc conditionnellement indépendantes sachant Y.

Connexions divergentes (fourchettes) Supposons qu'un facteur de transcription Y active les gènes X et Z. Si la concentration de Y est inconnue, le fait de savoir si X est activé nous renseigne sur l'état de Z (les gènes X et Z étant co-régulés). X et Z sont donc dépendants. Par contre, si la concentration de Y est connue, alors les niveaux d'expression de Z et de X sont totalement déterminés par la concentration de Y. X et Z sont donc conditionnellement indépendants sachant Y.

Connexions convergentes (*V-structure*) Supposons que les produits de deux gènes X et Z régulent l'expression du gène Y. Ce cas de figure est plus surprenant. Si le niveau d'expression de Y est inconnu, alors connaître le niveau d'expression de X ne nous fournit aucune information quant à celui de Z : X et Z sont (marginalement) indépendants. Par contre, si le niveau d'expression de Y est connu, alors connaître l'état de X nous renseigne sur l'état probable de Y. Par exemple, supposons que Y s'exprime si au moins l'un de ses deux régulateurs est activé, savoir que Y est exprimé et que X n'est pas activé implique que Z soit activé. On dit que les variables X et Z sont conditionnellement dépendantes sachant Y.

4.1.3 Équivalence Markovienne et ordre topologique

Lorsque des réseaux Bayésiens encodent les mêmes indépendances conditionnelles, il est impossible de les distinguer en se basant sur des données.

Exemple 4.1
Supposons que des tests d'indépendances conditionnelles nous indiquent que trois variables X, Y, Z sont telles que : $X \perp Z \mid Y$. Il est possible de modéliser cette situation au moyen de trois réseaux bayésiens différents : $X \leftarrow Y \leftarrow Z$, $X \rightarrow Y \rightarrow Z$, et $X \leftarrow Y \rightarrow Z$. On dit que ces trois modèles sont Markov-équivalents *et appartiennent à la même classe d'équivalence.*

Définition 4.2 (Équivalence Markovienne)
Des réseaux Bayésiens sont Markov-équivalents *lorsqu'ils encodent le même modèle statistique. Ils partagent alors le même graphe non orienté (squelette) ainsi que les mêmes structures convergentes (V-structures). L'orientation d'un arc n'appartenant pas à une V-structure peut varier d'un modèle à l'autre, à moins que l'inversion de cet arc n'introduise une nouvelle V-structure.*

Définition 4.3 (Classe d'équivalence)
Les modèles Markov-équivalents forment une classe d'équivalence. *Celle-ci est représentée de manière compacte par un* graphe sans cycle partiellement orienté *(ou PDAG pour partially directed acyclic graph) dont la topologie est définie par le squelette ainsi que les V-structures partagées par tous les modèles appartenant à cette classe d'équivalence. Pour s'assurer que l'inversion d'un arc n'appartenant pas à une V-structure existante n'introduise pas une nouvelle V-structure, l'orientation d'un tel arc peut être fixée. On obtient ainsi un* graphe sans cycle partiellement orienté complet *(ou CPDAG pour completed PDAG) représentant de manière adéquate une classe d'équivalence.*

Ces deux notions sont illustrées à la figure 4.3.

Il est important de noter que la notion d'équivalence Markovienne pose une limite théorique à l'apprentissage de structures à partir de données : quelle que soit la quantité d'observations disponible, même si l'on parvient à capturer toutes les indépendances conditionnelles entre les variables du système observé, il est impossible de trancher entre les différents modèles d'une classe d'équivalence. Il est seulement possible d'apprendre le CPDAG caractérisant cette classe d'équivalence.
Sur le plan biologique, cela signifie que l'on peut théoriquement identifier les interactions directes entre gènes à partir de données d'expression, mais qu'il est impossible pour nombre d'entre elles de savoir quel est le gène cible et quel est le régulateur. Bien sûr, si on dispose de données d'intervention, il est possible d'orienter un certain nombre d'arcs et de restreindre l'ensemble des modèles possibles.
Comme nous le verrons par la suite, il est courant de forcer l'orientation des arcs d'un réseau Bayésien en imposant un ordonnancement topologique sur ces sommets :

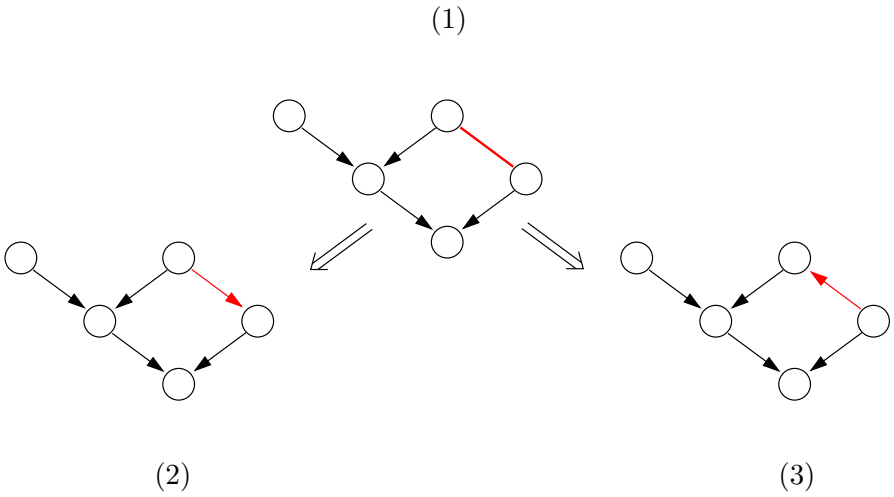


Fig. 4.3 – Illustration de l'équivalence Markovienne. (1) Un graphe sans cycle partiellement orienté représentant une classe d'équivalence. (2)(3) Deux graphes orientés sans cycle Markov-équivalents, appartenant à la classe d'équivalence représentée au-dessus.

Définition 4.4 (Ordre topologique)
Soit $\mathbf{X} = \{X_1, \ldots, X_n\}$ *un ensemble de sommets. Soient* $r_1, \ldots, r_n \in \mathbb{N}$. *Une application* $\pi : \{X_1, \ldots, X_n\} \mapsto \{r_1, \ldots, r_n\}$ *est un ordre topologique si et seulement si pour tout couple de variables* $(X_i, X_j) \in \mathbf{X} \times \mathbf{X}$ *de rangs respectifs* $\pi(X_i) = r_i$ *et* $\pi(X_j) = r_j$ *tel qu'il existe un chemin menant de* X_i *à* X_j, $r_i > r_j$.

De nombreux algorithmes utilisent cette information *a priori* pour simplifier l'apprentissage de la structure des réseaux Bayésiens. Nous verrons également que certaines stratégies visent à acquérir cette information à partir des données.

4.2 Apprentissage de paramètres dans les réseaux Bayésiens

La première étape dans l'apprentissage d'un réseau Bayésien est l'estimation de ses paramètres. L'approche la plus simple pour mener à bien cette tâche consiste à trouver les paramètres maximisant la probabilité pour que des données aient été générées par un modèle dont la structure est *fixée*. On parle alors d'apprentissage par maximum de vraisemblance. Il est également possible d'envisager une approche Bayésienne dans laquelle les paramètres sont considérés comme des variables aléatoires. On cherche alors à maximiser la probabilité *a posteriori* de ces paramètres (toujours à structure fixée) sachant les données. Cette approche permet d'introduire une distribution *a priori* sur les paramètres. Nous présentons ici le principe de ces deux approches. Le détail des calculs qui dépend du type de modèle local utilisé sera présenté sur un exemple précis dans la section 4.3.3.1.

4.2.1 Apprentissage par maximum de vraisemblance

Nous avons vu qu'un réseau Bayésien (RB) modélise la loi jointe $P(\mathbf{X})$. Il est possible d'apprendre les paramètres de ce modèle en recherchant le jeu de paramètres maximisant la probabilité pour que les observations soient des échantillons de ce modèle. On considère un ensemble

de données d'apprentissage $D = \{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$ comprenant m observations des n variables du modèle $\mathbf{X} = \{X_1, \ldots, X_n\}$. Pour fixer les idées, précisons qu'une observation consiste en une réalisation de ces n variables aléatoires : pour $i \in \{1, \ldots, m\}$, $\mathbf{x}^i = \{x_1, \ldots, x_n\}$. Nous pouvons à présent définir *la vraisemblance* du modèle, $L(\Theta)$, comme la probabilité d'observer les données sachant le modèle (c'est-à-dire sachant qu'elles suivent la loi jointe représentée par le modèle) :

$$L(\Theta) = P(D \mid \Theta) = \prod_{i=1}^{m} P(\mathbf{x}^i \mid \Theta) \tag{4.3}$$

Pour aboutir à cette factorisation, on fait l'hypothèse (très classique en apprentissage) que les données sont indépendantes et identiquement distribuées.

Le principe visant à identifier les paramètres Θ_{MV} maximisant $L(\Theta)$ est appelé *maximum de vraisemblance* (MV). Dans la pratique, on cherchera généralement à identifier les paramètres maximisant la log-vraisemblance $\ln P(D \mid \Theta)$, soit : $\Theta_{MV} = \arg\max_\Theta \ \ln P(D \mid \Theta)$. L'utilisation de la log-vraisemblance permet de simplifier les calculs et se justifie par le fait que la fonction ln est une fonction convexe (et donc $\ln(f)$ a les mêmes extrema que f). Il arrive également, que l'on préfère exprimer le problème sous la forme d'une minimisation de l'opposé de la log-vraisemblance. Enfin, notons que cette approche ne fait pas intervenir d'*a priori* sur les paramètres étudiés.

4.2.2 L'approche *Bayésienne*

Dans cette approche, on introduit une distribution *a priori* sur les paramètres. Celle-ci permet par exemple d'exploiter des connaissances (indépendantes des données d'apprentissage) concernant le phénomène modélisé afin de guider la recherche des paramètres. Lorsqu'on ne dispose d'aucune hypothèse ou connaissance a priori, il est courant d'utiliser une distribution a priori uniforme.
L'approche Bayésienne consiste à rechercher les paramètres maximisant la probabilité *a posteriori* (MAP) des données. On cherche à identifier Θ_{MAP}, le jeu de paramètres maximisant la probabilité d'un modèle candidat, sachant les données qu'il est censé représenter, soit : $\Theta_{MAP} = \arg\max_\Theta \ \ln P(\Theta \mid D)$.

En s'appuyant sur le théorème de Bayes, il est possible d'incorporer un *a priori* dans la détermination des paramètres :

$$P(\Theta \mid D) = \frac{P(D \mid \Theta) \cdot P(\Theta)}{P(D)}$$

On constate que la probabilité *a posteriori* s'exprime comme le produit d'une vraisemblance et d'une probabilité *a priori* ($P(\Theta)$), le tout divisé par une constante de normalisation. La probabilité des données $P(D)$ étant constante quel que soit le modèle traité, elle n'est pas prise en compte dans le calcul de ce critère.

4.3 Apprentissage de structure dans les réseaux Bayésiens

L'apprentissage de structure est l'aspect le plus intéressant des réseaux Bayésiens en biologie des systèmes. Il s'agit d'une tâche complexe qui nécessite le développement d'algorithmes adaptés et puissants. Elle est particulièrement difficile lorsque la quantité et la qualité des données disponibles est faible et que la conception des expériences de puces à ADN ne tient pas compte dès

le départ de certains impératifs de l'apprentissage statistique. Deux approches se partagent ce champ d'investigation : les approches *par contraintes* et la *recherche* de modèles fondée sur des approches à base de *score*, que nous avons privilégiées. Avant de présenter ces techniques nous allons d'abord rappeler les notations de base utilisées dans ce document. Surtout, nous rappellerons quelques principes et hypothèses (souvent implicites) sur lesquels se fondent l'inférence de modèles.

4.3.1 Problématique de l'apprentissage de réseaux de régulation et hypothèses de travail

Notations Dans ce qui suit, les variables aléatoires sont notées en majuscules $A, B, \ldots, Z$. Les valeurs prises par ces variables sont notées en minuscules $a, b, \ldots, z$. D'une manière générale, les ensembles sont notés en gras : qu'il s'agisse d'un ensemble de variables aléatoires $\mathbf{A} = \{B, C, D, E\}$ ou d'une instanciation de ce dernier $\mathbf{a} = \{b, c, d, e\}$.

Le problème d'apprentissage de structure Soit D un jeu de données correspondant à un ensemble d'observations indépendantes d'un système de régulation inconnu. On suppose que ce réseau de régulation peut être décrit par une distribution de probabilité jointe $P(\mathbf{X})$ sur une collection de variables aléatoires représentant les gènes d'intérêt $\{X_1, X_2, \ldots, X_n\}$. On suppose également que les observations expérimentales constituent un échantillon représentatif de cette loi jointe. On se donne une famille d'hypothèses correspondant à une classe de modèles probabilistes, en l'occurrence des réseaux Bayésiens $\mathcal{B} = (S, \Theta)$, qui nous semble adaptée à la représentation de la loi jointe. On souhaite identifier l'hypothèse (le modèle) représentant le plus fidèlement possible la loi jointe dont sont issues les données. Le modèle ainsi sélectionné doit permettre de décrire le système de régulation observé. Plus précisément, c'est la structure S de ce modèle qui nous intéresse, ses paramètres Θ étant secondaires dans la représentation du réseau de régulation. Le problème peut être posé d'une manière plus formelle de la manière suivante :
Soit $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ un ensemble de variables aléatoires dont nous souhaitons estimer la loi jointe $P(\mathbf{X})$. Nous modélisons celle-ci par un réseau Bayésien $\mathcal{B} = (S, \Theta)$ où S est la structure du modèle et Θ l'ensemble des paramètres de ce modèle. S est un DAG $S = \{\mathbf{X}, A\}$ où $\mathbf{X}$ est l'ensemble des sommets du DAG représentant les variables aléatoires d'intérêt et $A \subseteq \mathbf{X} \times \mathbf{X}$ l'ensemble des arcs du graphe codant les indépendances conditionnelles entre les variables. Dans le cas qui nous intéresse, les sommets du DAG correspondent aux niveaux d'expression d'un ensemble de gènes mesurés à travers diverses conditions expérimentales. L'ensemble de ces observations constitue un jeu de données noté D. $D = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ est un m échantillon de $P(\mathbf{X})$ tel que pour tout $j \in \{1, \ldots, m\}$, $\mathbf{x}^j = \{x_1^j, x_2^j, \ldots, x_n^m\}$ où x_i^j est la valeur prise par la variable X_i au sein de $\mathbf{x}^j$, la j^{e} observation parmi les m que compte la base d'exemples. Nous supposons que ces m observations sont indépendantes et identiquement distribuées. Nous souhaitons utiliser cette base d'exemples pour inférer la structure S du modèle $\mathcal{B}$ représentant la loi jointe des données.

Les hypothèses de travail Nous résumons à présent les principales hypothèses qui sous-tendent l'apprentissage de réseaux Bayésiens.

Définition 4.5 (Hypothèse de fidélité causale)
On postule l'existence d'un réseau Bayésien sur $\mathbf{X}$ *capable de représenter la liste des indépendances conditionnelles associée à la distribution de probabilité* $P(\mathbf{X})$ *sous-jacente aux données.*

L'hypothèse de fidélité causale revient à supposer que le modèle que nous nous proposons d'identifier existe bel et bien, et qu'il représente fidèlement la distribution des variables observées. Elle implique également que les données soient fiables, c'est-à-dire qu'il n'y a pas d'indépendances conditionnelles accidentelles.
Évidemment, les réseaux Bayésiens ne fournissent qu'une représentation abstraite et simplifiée des réseaux de régulation biologique. Comme nous l'avons expliqué dans le premier chapitre, l'étude du transcriptome ne permet de capturer qu'un aspect particulier de la régulation car les nombreux phénomènes régulatoires postérieurs à la synthèse des ARNm ne sont pas modélisés. Surtout, les réseaux Bayésiens (statiques) ne permettent pas de rendre compte du caractère dynamique des réseaux de régulation. On fait malgré tout l'hypothèse qu'il est possible de capturer la structure du système régulatoire à partir des données.

Définition 4.6 (Postulat causal de Markov)
Un ensemble de variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ *est* suffisant causalement *pour un jeu de données* D *si toute cause commune* Y *à un ensemble de variables de* $\mathbf{X}$ *appartient elle-même à* $\mathbf{X}$*, ou si* Y *est constante sur* D*. Cela implique que* $\mathbf{X}$ *est suffisant pour capturer toutes les relations d'indépendances conditionnelles pouvant être extraites des données. Relativement à cette hypothèse, chaque sommet est indépendant de ses non descendants conditionnellement à ses parents au sein du graphe.*

On suppose ici que toutes les variables du problème ont été prises en compte, et que par conséquent, toutes les indépendances conditionnelles caractérisant la distribution des données pourrons être extraites. Rappelons que dans le cas où une variable cachée intervient dans les phénomènes observés, il est possible de conclure à un lien de cause à effet entre deux phénomènes corrélés alors qu'ils ont en réalité une cause commune.
Dans le contexte des réseaux de régulation génétique, ce dernier point est également discutable. La technologie des puces à ADN permet de rendre compte de l'activité transcriptionnelle de l'ensemble des gènes d'un organisme (plusieurs dizaines de milliers) alors que les problèmes d'apprentissage à notre portée comportent de l'ordre de plusieurs dizaines de variables. Dans ces conditions, il apparaît évident que même un choix judicieux des gènes étudiés ne saurait garantir l'absence de variables cachées. Faute de solution satisfaisante, nous faisons cependant l'hypothèse que de telles variables, si elles existent, n'interviennent pas directement sur le système étudié ou ont un effet constant sur ce dernier.

4.3.2 Apprentissage par contraintes

Nous avons vu que dans les réseaux d'association et les modèles graphiques Gaussiens, la détermination des indépendances conditionnelles est utilisée pour décider si les données justifient l'incorporation d'une arête au sein du réseau. Le même principe permet de générer la structure du réseau Bayésien.

4.3.2.1 Test statistique d'indépendance conditionnelle

Le test statistique d'indépendance conditionnelle classiquement utilisé pour éliminer des arêtes surnuméraires ou ajouter des arêtes manquantes dans un réseau Bayésien est le test du χ^2 :

Définition 4.7 (Test du χ^2)
Soient deux variables aléatoires discrètes X_i et X_j appartenant à $\mathbf{X}$, qui prennent leurs valeurs respectivement dans $\{1,\dots,r_i\}$ et $\{1,\dots,r_j\}$. Soit N_{kl} le nombre de co-occurrences de $\{X_i = k, X_j = l\}$ dans la base d'apprentissage D de taille m, $N_{k\cdot}$ le nombre d'occurrences de $\{X_i = k\}$ et $N_{\cdot l}$ le nombre d'occurrences de $\{X_j = l\}$. Il s'agit de confronter le modèle observé dans les données $P_O = P(X_i, X_j)$ représenté par les occurrences $O_{kl} = N_{kl}$, au modèle théorique $P_t = P(X_i) \cdot P(X_j)$ représenté par les occurrences $T_{kl} = \frac{N_{k\cdot} \cdot N_{\cdot l}}{m}$. On considère la statistique suivante (de degrés de liberté $dl = (r_i - 1) \cdot (r_j - 1)$) :

$$\chi^2 = \sum_{k=1}^{r_i} \sum_{l=1}^{r_j} \frac{(O_{kl} - T_{kl})^2}{T_{kl}} = \sum_{k=1}^{r_i} \sum_{l=1}^{r_j} \frac{(N_{kl} - \frac{N_{k\cdot} \cdot N_{\cdot l}}{m})^2}{\frac{N_{k\cdot} \cdot N_{\cdot l}}{m}} \tag{4.4}$$

Sous l'hypothèse H_0, X_i et X_j sont indépendantes ce qui veut dire que $P(X_i, X_j) = P(X_i) \cdot P(X_j)$. Le test du χ^2 estime la plausibilité de l'hypothèse selon laquelle le modèle observé P_O correspond au modèle théorique P_t qui modélise l'hypothèse d'indépendance des variables. La valeur de la statistique de test est d'autant plus faible que les effectifs correspondant à ces deux modèles sont proches. L'hypothèse d'indépendance entre X_i et X_j est vérifiée si et seulement si $\chi^2 < \chi^2_{théorique}(dl, 1 - \alpha)$ pour un seuil de confiance α.

Ce test permet de rendre compte de l'indépendance de deux variables. Pour construire un réseau Bayésien, nous avons également besoin de tester l'indépendance de deux variables X_i et X_j conditionnellement à un ensemble de variables $\mathbf{X}_C$ dont les différentes configurations appartiennent à $\{1,\dots,r_C\}$. Pour cela, il suffit d'appliquer la même méthode pour confronter le modèle observé $P_O = P(X_i, X_j \mid \mathbf{X}_C)$ représenté par les occurrences $O_{klp} = N_{klp}$ (où N_{klp} est le nombre d'occurrences de $\{X_i = k, X_j = l, \mathbf{X}_C = p\}$) au modèle théorique $P_t = P(X_i \mid \mathbf{X}_C) \cdot (X_j \mid \mathbf{X}_C)$ représenté par les occurrences $T_{klp} = \frac{N_{k\cdot p} \cdot N_{\cdot lp}}{N_{\cdot\cdot p}}$ On considère alors la statistique suivante (de degrés de liberté $dl = (r_i - 1) \cdot (r_j - 1) \cdot r_C$) :

$$\chi^2 = \sum_{k=1}^{r_i} \sum_{l=1}^{r_j} \sum_{l=1}^{r_C} \frac{(O_{klp} - T_{klp})^2}{T_{klp}} \tag{4.5}$$

L'hypothèse d'indépendance entre X_i et X_j conditionnellement à $\mathbf{X}_C$ est vérifiée si et seulement si $\chi^2 < \chi^2_{\text{théorique}}(dl, 1 - \alpha)$ pour un seuil de confiance α.

Comme nous l'avons déjà mentionné dans le cas des modèles non orientés, lorsque le nombre de variables est important et que l'ensemble de conditionnement $\mathbf{X}_C$ devient grand, ce type d'approche devient inapplicable. En effet, la somme sur toutes les configurations de $\mathbf{X}_C$ dans l'équation (4.5) devient alors difficilement calculable et, à moins de disposer d'une base d'apprentissage très importante, les effectifs des configurations $\{X_i = k, X_j = l, \mathbf{X}_C = p\}$ sont alors trop faibles pour que le test soit concluant. Pour contourner ce problème, lorsque le nombre de données n'est pas assez important par rapport aux degrés de liberté, Spirtes et col. ont proposé de rejeter automatiquement l'hypothèse d'indépendance entre les variables X_i et X_j et de conclure à leur dépendance conditionnelle.

Différents travaux ont permis d'exploiter ces tests d'indépendance conditionnelle pour construire des réseaux Bayésiens. Les deux méthodes les plus célèbres à ce titre sont l'*algorithme IC* de Pearl et Verma et l'*algorithme PC* de Spirtes, Glymour et Scheines, dont nous rappelons les principes dans ce qui suit.

4.3.2.2 Les algorithmes PC et IC

Spirtes, Glymour et Scheines [SGS+00] ont proposé l'algorithme PC [4] pour reconstruire des réseaux Bayésiens à partir de tests d'indépendance conditionnelle. Celui-ci part d'un graphe non orienté complètement relié et teste les indépendances conditionnelles pour supprimer des arêtes. L'algorithme IC (pour *Inductive Causation*) [Pea00], introduit par Pearl à la même époque que l'algorithme PC, repose sur le même principe mais construit le graphe orienté par ajout successif d'arêtes en partant d'un graphe initialement vide.

Pour éviter d'avoir à tester toutes les indépendances conditionnelles possibles, dont le nombre augmente de manière exponentielle avec le nombre de variables, Spirtes et col. ont proposé un processus itératif qui limite la taille des ensembles de conditionnement $\mathbf{X}_C$ à chaque étape. Dans un premier temps, on teste les indépendances marginales de toutes les paires de variables au moyen de l'équation (4.4). Cela revient à réaliser un test d'indépendance conditionnelle d'ordre 0 ($\mathbf{X}_C = \emptyset$). Une fois les arêtes séparant les sommets marginalement indépendants supprimées, les arêtes restantes sont soumises à des tests d'indépendance conditionnelle d'ordre 1. De nouveau, un certain nombre d'arêtes ayant été supprimées, seul le sous-ensemble des arêtes encore présent dans le graphe est soumis à des tests d'indépendance conditionnelle d'ordre 2. Le processus se poursuit jusqu'à ce qu'aucune indépendance conditionnelle ne puisse être envisagée dans le graphe non orienté résultant. Cette procédure permet de diminuer le nombre d'indépendances conditionnelles à tester à chaque étape. Comme nous l'avons précisé plus haut, si la taille de l'ensemble de conditionnement devient trop élevée par rapport au nombre de données, les hypothèses d'indépendance sont automatiquement rejetées. Une fois le graphe de départ élagué, on teste la *dépendance conditionnelle* des sommets non adjacents ayant un ensemble de voisins communs afin de diriger les arêtes des V-structures.

Exemple 4.2
Considérons le cas simple où l'on a $X_i - X_C - X_j$. Des tests antérieurs ayant permis de conclure à l'indépendance de X_i et X_j, nous testons à présent l'indépendance de X_i et X_j conditionnellement à X_C. Si X_i et X_j ne sont plus indépendants lorsque leur voisin commun X_C est pris en compte, on peut conclure à leur indépendance conditionnelle. Dans un réseau Bayésien, cela se traduit par une V-structure $X_i \rightarrow X_C \leftarrow X_j$.

À ce stade on a un graphe sans cycle partiellement orienté représentant une classe d'équivalence. Afin d'obtenir un DAG, l'algorithme propage les orientations des arcs obtenus sur les arêtes adjacentes en prenant garde à ne pas introduire de nouvelles V-structures. Cependant, à moins de disposer d'un ordre topologique des sommets du graphe permettant d'orienter les arcs de manière univoque, cette méthode ne permet pas de générer un DAG unique.

Il existe différentes variantes à l'algorithme PC visant à optimiser la phase d'élagage du graphe en limitant le nombre et l'ordre des tests d'indépendance conditionnelle à réaliser. L'algorithme PC* notamment, propose de limiter la construction d'un ensemble de conditionnement $\mathbf{X}_C$ aux variables adjacentes à X_i et X_j se trouvant sur un chemin entre X_i et X_j. Cette méthode est cependant lourde à mettre en œuvre dans la mesure où elle nécessite de stocker tous les chemins possibles dans le graphe.

Pour des variables discrètes ou normalement distribuées, si l'hypothèse de fidélité causale et le postulat causal de Markov (voir section 4.3.1) sont respectés, l'algorithme PC converge vers

[4] Cet algorithme doit sa dénomination aux initiales des prénoms de deux de ses concepteurs : Peter (Spirtes) et Clark (Glymour).

la solution exacte dans la limite des grands échantillons. Par ailleurs, si on fixe un nombre de parents maximum pour chaque variable, la complexité de l'algorithme devient polynomiale par rapport au nombre de variables. Dans les faits cependant, lorsque le nombre de variables est important, ces approches deviennent vite difficile à mettre en œuvre.

4.3.3 Apprentissage par exploration d'un espace de recherche

Pour des applications en biologie des systèmes, l'apprentissage de structure à base de scores est généralement privilégiée. Son principe est simple : il s'agit de trouver parmi tous les modèles possibles celui dont la structure explique le mieux les observations. On considère donc l'espace de recherche [5] constitué de l'ensemble des DAG ayant les variables de $\mathbf{X}$ pour sommets. On se donne un critère de qualité généralement appelé *score* permettant d'évaluer chacun de ces modèles [6]. Pour l'essentiel, ce critère représente la capacité d'un modèle candidat à rendre compte des données. Afin de trouver la structure maximisant ce critère, il est nécessaire de définir une méthode permettant de parcourir judicieusement l'espace des DAG candidats (la plus simple consistant à évaluer toutes les possibilités). Dans un premier temps nous allons définir les scores permettant d'évaluer des solutions candidates et présenter quelques unes de leurs propriétés les plus importantes. Dans un second temps, nous présenterons les principales méthodes de recherche du DAG optimal. Une description plus exhaustive et plus détaillée que celle que nous fournissons de ces différentes approches est disponible dans le mémoire de Philippe Leray [Ler06].

4.3.3.1 Évaluation d'une structure candidate

Nous allons à présent présenter les quatre approches les plus couramment utilisées dans la littérature pour évaluer une structure candidate : la vraisemblance, le critère d'information Bayésien, l'approche Bayésienne et le principe de la longueur de description minimum. Les trois premières sont des extensions des critères utilisés pour apprendre les paramètres d'un réseau Bayésien, alors que la quatrième est inspirée de la théorie du codage et de la théorie de de l'information. Comme nous avons fait l'hypothèse que les données traitées sont discrètes, les équations présentées par la suite s'appliquent à des modèles discrets. Cependant, il est relativement aisé de dériver des variantes de ces équations pour des modèles continus basés sur des CPD simples telles que les distributions Gaussiennes linéaires. Avant toute chose, nous allons évoquer certaines propriétés communes à la plupart des scores présentés par la suite.

Propriétés générales des scores Les méthodes d'apprentissage de structure à base de scores reposent le plus souvent sur une heuristique de recherche au sein de l'espace des DAG. Le parcours au sein de cet espace étant le plus souvent réalisé au moyen d'opérateurs d'ajout ou de suppression d'arcs, il est avantageux de disposer d'un score calculable localement. Il est ainsi possible de calculer la variation de ce score en fonction de la contribution de l'arc ajouté ou enlevé, sans qu'il faille ré-évaluer le DAG dans sa globalité. C'est là qu'intervient la notion de score décomposable.

Définition 4.8 (Score décomposable)
Un score permettant d'évaluer globalement un réseau Bayésien $\mathcal{B}$ est dit décomposable s'il peut être écrit comme une somme (ou un produit) de n mesures $s(X_i, Pa_i)$ (fréquemment appelées

[5] De manière générale il s'agit de l'ensemble des solutions admissibles pour un problème d'optimisation. Il correspond à l'espace de définition de la fonction à maximiser, à moins que des contraintes externes ne restreignent le nombre de solutions prises en compte.

[6] Par abus de langage, nous parlerons souvent de « modèle » au lieu de « structure de modèle »

score locaux*), telles que chacune n'est fonction que d'un sommet X_i et de ses parents Pa_i au sein du graphe :*

$$Score(\mathcal{B}) = \sum_{i=1}^{n} s(X_i, Pa_i) \quad \text{ou} \quad Score(\mathcal{B}) = \prod_{i=1}^{n} s(X_i, Pa_i) \tag{4.6}$$

Comme nous l'avons vu précédemment, plusieurs DAG distincts peuvent encoder le même ensemble d'indépendances conditionnelles. En l'absence d'informations *a priori* concernant la topologie de la structure objectif, il n'y a aucune raison de privilégier l'un d'entre eux lors de l'apprentissage. Dans ce cas, il est pertinent d'associer la même valeur de score à des structures appartenant à la même classe d'équivalence Markovienne. On souhaite donc disposer de *scores Markov-équivalents*.

Définition 4.9 (Score Markov-équivalent)
Un score qui associe une même valeur à deux graphes Markov-équivalents est dit Markov-équivalent.

Les scores que nous allons présenter par la suite sont asymptotiquement consistants. Cela signifie que pour de grands échantillons, aucun DAG ne reçoit une valeur de score supérieure à celle du DAG objectif. En général, si l'ordonnancement topologique des variables est connu, alors le DAG maximisant l'un de ces scores est unique. Cependant, dans le cadre que nous nous sommes fixé (apprentissage de structure à partir d'une collection raisonnablement grande de données statiques) aucune de ces conditions n'est remplie : la quantité de données est relativement faible au regard des considérations asymptotiques évoquées plus haut et dans les faits, l'ordonnancement topologique, qui est une connaissance *a priori* difficilement accessible, n'est pas utilisé.

Maximum de vraisemblance et sur-apprentissage L'adéquation de la structure S d'un modèle probabiliste aux données peut être mesurée par sa vraisemblance $P(D \mid S, \widehat{\Theta_S})$. Les paramètres $\widehat{\Theta_S}$, sont eux-mêmes estimés par maximum de vraisemblance. La vraisemblance peut donc être utilisée pour évaluer une structure candidate et constitue à ce titre un score. L'approche consistant à trouver la structure maximisant la vraisemblance (c'est-à-dire celle dont les paramètres maximisent la vraisemblance pour tous les modèles possibles) est l'apprentissage par maximum de vraisemblance. Il est utile de rappeler que le nombre et la nature des paramètres d'un modèle sont totalement déterminés par la structure du modèle, celle-ci définissant la forme des CPD que ces paramètres caractérisent. Un score pour une structure candidate S est alors donné par :

$$L(S) = \max_{\Theta_S} P(D \mid S, \Theta_S) \tag{4.7}$$

L'hypothèse selon laquelle les observations de la base d'apprentissage sont indépendantes et identiquement distribuées nous permet de factoriser la vraisemblance de la base d'apprentissage comme le produit des vraisemblances de chacune des m observations :

$$P(D \mid S, \Theta_S) = \prod_{j=1}^{m} P(\mathbf{x}^j \mid S, \Theta_S) \tag{4.8}$$

En exploitant la structure du modèle, il est de nouveau possible de factoriser la vraisemblance d'une observation x^j de D suivant la formule (4.2) décrivant la décomposition de la loi jointe

dans un réseau Bayésien :

$$P(\mathbf{x}^j \mid S, \Theta_S) = \prod_i P(x_i^j \mid pa_i^j) \tag{4.9}$$

où x_i^j et pa_i^j sont respectivement la j^{e} observation de la variable X_i et de ses parents Pa_i au sein du graphe S.
Pour des modèles discrets dont les CPD sont représentées par des tables de probabilité conditionnelle, la probabilité conditionnelle pour qu'une variable X_i prenne la valeur k sachant que ses parents Pa_i sont dans la configuration l est donnée par le paramètre θ_{ik}^l. Pour toute observation de la base d'apprentissage on a donc :

$$P(X_i = k \mid Pa_i = l) = \theta_{ik}^l \tag{4.10}$$

Pour un modèle complet (dont la structure et les paramètres sont fixés), il est donc possible d'exprimer simplement la vraisemblance des données de la manière suivante :

$$P(D \mid S, \Theta_S) = \prod_i \prod_k \prod_l {\theta_{ik}^l}^{N_{ik}^l} \tag{4.11}$$

où l'exposant N_{ik}^l est le nombre de co-occurrences de $X_i = k$ et $Pa_i = l$ dans les données. Il est courant de calculer la log-vraisemblance des données qui donne lieu à une expression plus simple à manipuler :

$$\log P(D \mid S, \theta) = \sum_i \sum_k \sum_l N_{ik}^l \cdot \log(\theta_{ik}^l) \tag{4.12}$$

Dans le cadre discret, pour une structure de modèle fixée, le paramétrage $\widehat{\Theta_S} = \arg\max_{\Theta_S} P(D \mid S, \Theta_S)$ maximisant la vraisemblance des données repose sur l'ensemble des paramètres $\hat{\theta}_{ik}^l$ tels que pour toute variable X_i ainsi que pour toute valeur k prise par cette variable et toute configuration l prise par ses parents :

$$\hat{\theta}_{ik}^l = \frac{N_{ik}^l}{N_i^l} \tag{4.13}$$

où $N_i^l = \sum_k N_{ik}^l$ est le nombre de fois où les parents de X_i prennent la configuration de valeur l dans les données.
Chaque paramètre $\hat{\theta}_{ik}^l$ est l'estimation par maximum de vraisemblance de θ_{ik}^l. Dans la mesure où ces estimations reposent sur de simples calculs de fréquences, le temps nécessaire pour calculer la vraisemblance d'une structure candidate est relativement faible. Cette caractéristique est fondamentale car comme nous le verrons dans la partie dédiée aux résultats numériques, l'évaluation d'une structure candidate est l'opération la plus gourmande dans un algorithme d'apprentissage de structure. Dans la mesure où elle doit être répétée un grand nombre de fois durant le parcours de l'espace des DAG, une procédure d'évaluation trop complexe et trop coûteuse est rédhibitoire lorsque cet espace est large (ce qui est toujours le cas).
Au final, la vraisemblance peut être avantageusement remplacé par la log-vraisemblance. Dans le cas discret, on peut donc évaluer l'adéquation d'une structure candidate S aux données D grâce à la formule suivante :

$$Score_{MLV}(S) = \sum_i \sum_k \sum_l N_{ik}^l \cdot \log(\hat{\theta}_{ik}^l) \tag{4.14}$$

Bien que très simple à mettre en œuvre, cette approche est inappropriée dans la mesure où elle est sujette au sur-apprentissage. Ce concept (classique en apprentissage) désigne le fait qu'un modèle trop riche est susceptible de décrire parfaitement les données d'apprentissage sans pouvoir pour autant rendre compte d'un nouveau jeu d'observations portant pourtant sur le même phénomène. En somme, le modèle appris décrit spécifiquement les données d'apprentissage mais ne représente pas le phénomène observé. Pour lutter contre le sur-apprentissage il est classique d'appliquer le principe du rasoir d'Occam. Ce dernier avance que parmi plusieurs explications possibles, la plus simple doit être préférée. Dans le cas d'un réseau Bayésien, un graphe complexe ayant une connectivité très élevée peut théoriquement représenter n'importe quelle loi jointe. Il suffit pour cela d'adapter les paramètres des CPD pour faire en sorte qu'un arc surnuméraire n'influe pas sur la distribution de sa cible. Pour une CPD Gaussienne linéaire, cela se caractériserait par un poids faible ou nul sur l'état du parent en question. L'approche du maximum de vraisemblance, en l'état, ne permet pas d'éviter le piège du sur-apprentissage et est susceptible de générer systématiquement des modèles trop riches, comportant trop d'arcs. La solution standard pour échapper à ce problème consiste à pénaliser les modèles complexes, de sorte que pour une valeur de vraisemblance similaire, le modèle le plus parcimonieux (comportant moins d'arcs et donc moins de paramètres) est préféré.

Le critère d'information Bayésien Avant tout chose, il faut préciser que contrairement à ce que son nom laisse présager, le critère d'information Bayésien (BIC) n'est pas un score Bayésien. Il a été défini par Schwarz en 1978 [Sch78] comme un moyen général d'estimer la complexité d'un modèle statistique. Ce score permet de contrôler le sur-apprentissage en pénalisant la vraisemblance du modèle en fonction de sa complexité, caractérisée par le nombre de paramètres. Le BIC est défini par :

$$Score_{BIC}(S) = \max_{\Theta_S} \log(P(D \mid S, \Theta_S)) - \frac{Dim(\mathcal{B})}{2} \log(m) \tag{4.15}$$

où $Dim(\mathcal{B})$ est la dimension (le nombre de paramètres) du modèle $\mathcal{B} = (S, \Theta)$ et m est le nombre de données. Le facteur $\log(m)$ permet donc de déterminer l'importance relative accordée à la pénalisation par rapport au terme de vraisemblance, en fonction de la taille de la base d'apprentissage.

La dimension du modèle probabiliste est égale à la somme des dimensions de ses composantes, en l'occurrence les CPD qui sont les facteurs de la loi jointe encodée par le réseau Bayésien $\mathcal{B}$:

$$Dim(\mathcal{B}) = \sum_{i=1}^{n} Dim(\mathcal{B}^i) \tag{4.16}$$

où $Dim(\mathcal{B}^i)$ est la dimension de la i^{e} composante de $\mathcal{B}$. Dans un modèle discret, $Dim(\mathcal{B}^i)$ est égale au nombre d'éléments de la table de probabilités conditionnelles décrivant $P(X_i \mid Pa_i)$. Le nombre de ces éléments peut être déterminé de la manière suivante : soit r_i l'arité de la variable X_i et q_i le nombre de configurations possibles des parents de X_i défini par $q_i = \prod_{X_j \in Pa_i} r_j$. Le nombre de paramètres nécessaires à la représentation de $P(X_i \mid Pa_i = pa_i)$ étant égale à $r_i - 1$, le nombre de paramètres requis pour décrire $P(X_i \mid P_i)$ est égal à $Dim(\mathcal{B}^i) = (r_i - 1) \cdot q_i$. Par conséquent le nombre de paramètres du modèle, utilisé pour évaluer sa complexité, est défini de la manière suivante :

$$Dim(\mathcal{B}) = \sum_{i=1}^{n} (r_i - 1) \cdot q_i \tag{4.17}$$

En reprenant l'équation (4.14) de la vraisemblance d'une structure S ainsi que la formule (4.17) permettant d'évaluer sa complexité nous pouvons réécrire le score BIC de la façon suivante :

$$Score_{BIC}(S) = \sum_i \sum_k \sum_l N_{ik}^l \cdot \log(\hat{\theta}_{ik}^l) - \frac{(r_i - 1) \cdot q_i)}{2} \log(m) \tag{4.18}$$

où $\hat{\theta}_{ik}^l = \frac{N_{ik}^l}{N_i^l}$ avec $N_i^l = \sum_k N_{ik}^l$. Notons qu'il est courant d'exprimer le score BIC non pas comme un critère de qualité à maximiser mais comme une fonction de coût à minimiser :

$$Score_{BIC}(S) = \sum_i \sum_k \sum_l -2 \cdot N_{ik}^l \cdot \log(\hat{\theta}_{ik}^l) + (r_i - 1) \cdot q_i \cdot \log(m) \tag{4.19}$$

Ce changement mineur n'entraîne cependant aucune modification dans les propriétés du score BIC.

Au même titre que le score de la longueur minimum de description que nous verrons par la suite, le score BIC permet de maximiser la précision du modèle tout en pénalisant sa complexité. Il est également Markov-équivalent, deux DAGs équivalents au sens de Markov ayant la même log-vraisemblance (car ils encodent les mêmes indépendances conditionnelles) et la même complexité [Chi95]. Enfin, le score BIC est un score décomposable, qui s'exprime comme la somme de scores locaux caractérisant chacun l'une des familles composant le réseau.

L'approche du maximum *a posteriori* Lorsque l'on souhaite adopter une approche Bayésienne, on est amené à maximiser la probabilité *a posteriori* de la structure du modèle :

$$P(S \mid D) = \frac{P(D \mid S) \cdot P(S)}{P(D)} \tag{4.20}$$

où $P(D)$ est une constante de normalisation correspondant à la moyenne de la vraisemblance sur toutes les structures possibles. Dans la mesure où elle ne dépend pas du modèle considéré, elle est fréquemment éliminée du calcul de la probabilité *a posteriori* qui est utilisée pour évaluer les structures candidates d'un réseau Bayésien. À ce titre, la probabilité *a posteriori* constitue donc un score, appelé *score Bayésien*, que l'on cherche à maximiser. Le calcul de ce dernier repose sur la vraisemblance marginale des observations étant donnée la structure du modèle $P(D \mid S)$ et sur la probabilité *a priori* de cette structure $P(S)$. Nous allons à présent étudier tour à tour ces deux composantes.

La vraisemblance marginale Dans la majorité des cas, lorsque l'on ne dispose d'aucune connaissance *a priori* concernant le modèle recherché, on utilise un *a priori* $P(S)$ uniforme et le score Bayésien repose alors totalement sur la vraisemblance marginale de la structure du modèle. Le calcul de la vraisemblance marginale $P(D \mid S)$ constitue la principale difficulté dans le calcul d'un score Bayésien. Elle peut s'écrire de la manière suivante :

$$P(D \mid S) = \int_\Theta P(D \mid S, \theta) \cdot P(\theta \mid S) d\theta \tag{4.21}$$

où $P(D \mid S, \theta)$ est la vraisemblance des observations pour un modèle complet, et $P(\theta \mid S)$ la probabilité *a priori* des paramètres de ce modèle.
La différence fondamentale avec les scores précédents repose sur le traitement des paramètres

des CPD qui sont intégrés et non pas maximisés. Cette particularité permet d'éviter le sur-apprentissage auquel le critère de vraisemblance et le BIC (malgré sa contrainte de parcimonie) sont théoriquement exposés.
Dans le cas général, le calcul de cette intégrale est difficile à réaliser. Pour qu'il existe une solution analytique à cette équation, il est nécessaire que $P(D \mid S, \theta)$ et $P(\theta \mid S)$ soient conjuguées, c'est-à-dire que ces distributions appartiennent à la même famille de fonctions, de telle sorte que la distribution *a posteriori* $P(S \mid D)$ ait la même forme que la distribution *a priori* $P(\theta \mid S)$. Dans le cas contraire, il est nécessaire de trouver une solution approchée au calcul de la vraisemblance marginale.

Cooper et Herskovits [CH92] ont proposé une solution exacte pour le calcul de la vraisemblance marginale dans des réseaux Bayésiens discrets. Ils utilisent la distribution *a priori* conjuguée d'une CPD suivant une loi multinomiale, appelée « *a priori* de Dirichlet ». Les auteurs font l'hypothèse que les distributions *a priori* de Dirichlet sur les paramètres de chaque CPD sont indépendantes. D'une manière générale, ils supposent également qu'il n'y a ni données manquantes ni variables cachées et que les observations de la base d'apprentissage sont indépendantes et identiquement distribuées. Sous ces conditions, on peut décomposer le score Bayésien global en scores locaux caractérisant chacune des familles composant la structure du modèle. Il est alors possible de calculer le score Bayésien de Dirichlet (score BD).

Définition 4.10 (Score Bayésien de Dirichlet (BD))

$$Score_{BD}(S) = P(D \mid S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

où Γ est la loi Gamma, r_i est le nombre de modalités de la variable X_i et q_i est le nombre de configurations possibles de Pa_i, la parenté de X_i. N_{ijk} est le nombre de fois où $X_i = k$ et $Pa_i = j$ au sein de D et $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Enfin, $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, où les α_{ijk} sont les coefficients de Dirichlet. Ces derniers peuvent être vus comme un « comptage virtuel » du nombre de fois où $X_i = k$ et $Pa_i = j$ indépendamment de toute observation expérimentale. Ils traduisent un *a priori* sur les paramètres du modèle dans le sens où l'on augmente le nombre de comptages virtuels d'une configuration $(X_i = k, Pa_i = j)$ pour traduire le fait que le modélisateur croit en une valeur élevée du paramètre codant la probabilité de cette configuration $P(X_i = k, Pa_i = j)$. Lorsque l'on a aucun *a priori* sur la valeur de ces paramètres, la distribution de probabilité sur ces derniers est uniforme.
Heckerman et col. [HG95] ont montré que le score Bayésien de Dirichlet n'est pas Markov-équivalent. Ils ont donc proposé une variante de celui-ci, le score *Bayesian Dirichlet equivalent* (BDe), qui permet de corriger cet inconvénient.

Des résultats similaires existent dans le cas continu, pour des modèles Gaussiens utilisant un *a priori* conjugué normal de Wishart [GH97]. Il est également possible d'appliquer l'approche Bayésienne à des réseaux présentant des structures locales, c'est-à-dire dont les CPD sont représentées par des arbres [FGA99]. Dans ce cas de figure, la vraisemblance marginale au niveau de chacun des sommets du DAG se décompose de nouveau en composantes indépendantes pour chacune des feuilles de l'arbre de régression local.
Pour des modèles plus complexes il n'existe pas, à l'heure actuelle, de solution analytique permettant de calculer la vraisemblance marginale. Cette dernière doit donc être approchée au moyen de deux techniques communément employée pour l'approximation d'intégrale : les méthodes

d'échantillonnage tel que l'échantillonnage de Gibbs utilisé par Bulashevska et Eils [BE05] pour estimer les probabilités *a posteriori* de la structure $P(S \mid D)$ et des paramètres $P(\Theta \mid D)$ du modèle, et l'approximation de Laplace utilisée par Imoto et col. [IGM02].
Bien que le score BIC ne soit pas à proprement parler un score Bayésien, il peut être vu, sous certaines conditions, comme une approximation de ce dernier. Plus particulièrement, il est une estimation de la vraisemblance marginale de la structure du modèle obtenue par l'utilisation de l'approximation de Laplace [Raf95]. Lorsque la distribution *a priori* sur les structures du modèle est uniforme, le score BIC peut alors être vu comme une approximation de la probabilité *a posteriori* d'une structure S sachant D [NRF04]. L'approximation de Laplace peut s'avérer imprécise dans la mesure où l'une des principales hypothèses sur laquelle elle repose est l'existence d'une base d'apprentissage de grande taille, or nous avons vu que le nombre d'observations en réalité disponible est généralement restreint. Elle demeure cependant une approche intéressante, les méthodes d'échantillonnage ayant un coût de calcul très important. En effet, ce dernier point est souvent rédhibitoire dans la mesure où l'exploration de l'espace des structures peut s'avérer interminable si le calcul du score de chaque structure candidate met lui même en oeuvre un algorithme long et coûteux.

Utilisation d'un *a priori* sur les structures Le second élément nécessaire au calcul d'une probabilité *a posteriori* $P(S \mid D)$ est une distribution de probabilité *a priori* sur l'ensemble des structures possibles du modèle $P(S)$. Elle permet de guider le processus d'apprentissage vers des modèles plus réalistes en se fondant sur des connaissances *a priori* des experts en biologie, mais aussi en intégrant des informations obtenues à partir de sources de données alternatives. Le plus souvent, les connaissances concernant les systèmes biologiques étudiés sont encodées dans une réseau S^p (structure *a priori*) qui est raffiné par le processus d'apprentissage. La première idée consiste à restreindre l'espace des structures candidates à un voisinage $\mathcal{V}(S^p)$ du réseau *a priori* de telle sorte que tous les DAG appartenant à ce voisinage sont jugés équiprobables. Une manière de concevoir ce voisinage est de prohiber certains arcs ou au contraire, de les rendre obligatoires au sein de la structure recherchée. Les structures violant ces contraintes se voient attribuées une probabilité *a priori* nulle, alors que toutes celles qui les respectent et correspondent, sur ce point, au réseau *a priori*, sont équiprobables.
On peut ainsi définir une distribution *a priori* sur les structures du modèle de la manière suivante :

$$P(S) = 1/ \mid \mathcal{V}(S^p) \mid \text{ si } S \in \mathcal{V}(S^p), 0 \text{ sinon} \tag{4.22}$$

Cette distribution est particulièrement rigide. Elle implique que les connaissances qui s'y expriment ne souffrent aucune remise en question. Dans les faits, il est rare qu'un expert puisse garantir la présence ou l'absence d'un arc (ou de toute autre caractéristique topologique) dans un réseau. Généralement, il énonce des hypothèses et non des affirmations. C'est pourquoi il est utile de disposer de distributions *a priori* plus souples, capables d'intégrer l'incertitude relative aux informations fournies par les experts. Dans cette optique, on considère une mesure de confiance quant à la présence ou à l'absence d'un arc (i, j) dans le réseau, notée $0 < k_{i,j} < 1$. Un *a priori* sur une structure du modèle peut alors s'exprimer (à une constante de normalisation près) comme le produit des poids $k_{i,j}$ sur tous les arcs $(i, j) \in \mathbf{X} \times \mathbf{X}$:

$$P(S) \propto \prod_{(i,j) \in A} k_{i,j} \tag{4.23}$$

La constante de normalisation (qui est nécessaire pour obtenir une densité de probabilité) est identique pour tous les modèles considérés. Par conséquent, elle peut être mise de côté lorsque

l'on souhaite calculer des probabilités *a posteriori* relatives à différentes structures.
Bien que cette approche séduise par sa simplicité, elle est mathématiquement incorrecte. L'*a priori* que nous venons de décrire est défini sur l'espace des graphes orientés ayant les éléments de $\mathbf{X}$ comme sommets. En réalité, nous nous intéressons à l'espace des graphes *sans cycle* orientés. Par conséquent, la distribution *a posteriori* des structures d'un réseau Bayésien est calculée au moyen d'une distribution *a priori* définie sur un espace distinct : celui des graphes orientés (avec ou sans cycle). Ce problème est généralement contourné par la seule prise en compte des structures sans cycles. Par ailleurs, certains auteurs tels que Imoto et col. [IKG+03] font valoir que la formule (4.23) fournit une approximation satisfaisante de $P(S)$, suffisante pour distinguer les mérites de différents modèles candidats au regard des connaissances *a priori* disponibles.

Plusieurs solutions sont envisageables pour choisir les poids $k_{i,j}$ nécessaires à l'expression de cette probabilité *a priori*. Heckerman et col. [HGC95] ont proposé d'utiliser une constante de pénalisation $k_{i,j} = k$ pour tous les arcs qui varient entre une structure candidate S et la structure *a priori* S^p. Par conséquent $P(S) \propto k^{\varepsilon}$ où ε est le nombre d'arcs qui diffèrent entre S et S^p.

Imoto et col. [IKG+03] ont suggéré d'incorporer des informations relatives aux interactions protéine-protéine et protéine-ADN, ainsi qu'aux sites de fixation des régulateurs sur les gènes cibles, dans la pondération des différentes structures candidates. Plus spécifiquement, Tamada et col. [TKB+03] ont proposé de modifier de manière itérative la distribution *a priori* sur les modèles candidats en fonction de la mise en évidence de sites de fixation pour des facteurs de transcription durant l'apprentissage. À chaque étape de leur algorithme, un réseau Bayésien est construit à partir de données de puces à ADN. En se basant sur la structure de ce dernier, on souhaite identifier des facteurs de transcription putatifs susceptibles de réguler plusieurs gènes de concert. Pour chaque sommet X_i du réseau Bayésien courant, on recherche des séquences consensus dans les régions promotrices des enfants et des petits-enfants de ce sommet. Si de telles séquences sont identifiées, les sommets concernés (notés C_i) sont alors susceptibles d'être co-régulés par X_i. Il est en effet raisonnable d'envisager que X_i soit un facteur de transcription se fixant sur les motifs présents au sein des séquences régulatrices de ses proches descendants dans le DAG. Le réseau Bayésien est alors réappris en renforçant les poids des arcs allant de X_i vers les sommets de C_i, tandis que les autres arcs voient leur poids diminuer.
Cette approche implique que l'on dispose des séquences régulatrices des gènes représentés dans le réseau Bayésien. Cette contrainte n'est pas limitante, dans la mesure où les données de séquençage sont aujourd'hui relativement accessibles pour un nombre sans cesse croissant d'organismes. L'inconvénient majeur de cette méthode réside dans l'étape d'alignement de ces séquences, qui peut s'avérer très coûteuse en termes de temps de calcul, surtout si le nombre de gènes à prendre en compte (et donc le nombre d'alignements à réaliser) à chaque itération est important.

Bernard et col. [BH05] ont quant à eux choisi d'utiliser des données de *ChIP on chip* (voir section 2.2 page 22) afin de construire la distribution *a priori* sur leurs modèles. L'idée principale consiste à étudier la fixation d'une protéine (typiquement un facteur de transcription) sur les séquences régulatrices d'un ensemble de gènes au moyen d'une puce à ADN. Observer une telle fixation ne permet pas de préjuger de l'effet de cette protéine sur le gène ciblé, cependant cela permet d'attirer l'attention sur une possible régulation : si des données d'expression suggèrent que le produit d'un gène A régule un gène B, le fait de savoir que la protéine codée par A se fixe sur la séquence promotrice de B renforce considérablement la vraisemblance de cette régulation. Inversement, l'absence de fixation d'une protéine sur la séquence régulatrice d'un gène ne signi-

fie nullement que la première ne régule pas l'expression du second. Une expérience de *ChIP on chip* génère une collection de mesures d'intensités de fluorescence caractérisant la quantité de protéines fixée sur la séquence promotrice de chacun des gènes représentés sur la puce [7]. Pour chaque spot i, une p-valeur est calculée. Les auteurs font l'hypothèse que cette dernière suit une loi exponentielle lorsqu'un arc entre la protéine étudiée et le gène i est présent dans le modèle et qu'elle suit une loi uniforme lorsque cet arc est absent. Grâce à la règle de Bayses, ils expriment la probabilité pour qu'un arc soit présent sachant la p-valeur calculée à partir des données de fixation. Après avoir intégré les paramètres libres de cette distribution exponentielle, la probabilité finale est utilisée pour pondérer l'arc correspondant au sein de la distribution *a priori* de la structure.

A contrario, lorsque l'on ne dispose d'aucune information *a priori* permettant de guider l'apprentissage, toutes les structures candidates sont jugées équiprobables et $P(S)$ est donc uniforme. Dans l'immense majorité des cas où les données d'expression sont la seul source d'information exploitée pour apprendre le modèle, c'est cette approche qui est retenue.

Le principe de la longueur de description minimum (MDL) Les scores étudiés jusque là sont tous des scores probabilistes. Une autre approche fondée sur un paradigme différent, bien que proche dans son principe du score Bayésien [CS95], a été proposée par Lam et Bacchus [LB94] pour évaluer des modèles candidats. Il s'agit du *principe de la longueur de description minimum* (ou MDL pour *Minimum Description Length*) qui se fonde sur la théorie du codage. Une façon de voir le principe MDL est de le considérer comme une approche Bayésienne pour laquelle la distribution *a priori* sur l'espace des modèles diminue avec la longueur de leur codage. Cette dernière rendant compte de leur complexité, le principe MDL impose donc un biais en faveur des modèles les plus simples.
La perte de précision qui en résulte dans la représentation des données est compensée par une meilleure capacité de généralisation du modèle. Effectivement, comme nous l'avons vu avec le score BIC, contraindre la complexité d'un modèle est un moyen efficace de lutter contre le sur-apprentissage. De plus, des réseaux ayant une faible connectivité présentent l'avantage conceptuel d'être plus simples à comprendre.

Plus précisément, le principe MDL de Rissanen [Ris78] avance que le meilleur modèle pour une collection d'observations est celui qui *minimise* la somme de deux termes :

1. La longueur du codage des données décrites par le modèle, un bon modèle devant représenter les données de manière compacte. Il peut être utile de songer au modèle probabiliste comme un moyen de compresser les données.
2. La longueur de codage du modèle lui-même, ce dernier devant être représenté pour pouvoir être utilisé.

Nous allons commencer par présenter le type de codage permettant de représenter un modèle candidat. Dans un second temps, nous détaillerons le codage des données obtenu à partir d'un tel modèle.

Codage du modèle Pour représenter un réseau Bayésien, il est nécessaire et suffisant de disposer des informations suivantes :

- une liste des parents pour chaque sommet permettant de définir la structure du réseau ;
- l'ensemble des paramètres des distributions de probabilité conditionnelle associées à chaque sommet.

[7] Nous supposons que chaque spot de la puce correspond à la séquence promotrice d'un gène donné.

Nous nous intéressons d'abord au codage de la structure. Supposons que le réseau comprenne n sommets. Pour un sommet X_i ayant k_i parents, $k_i \cdot \log_2(n)$ bits seront nécessaires pour stocker la liste de ces parents. La longueur de codage d'un DAG est la somme de la longueur de codage des listes de parents qui le composent soit : $\sum_{i=1}^{n} k_i \cdot \log_2(n)$.

Nous considérons à présent le codage des paramètres. Dans le cas discret où les CPD sont des tables de probabilités conditionnelles, la taille du codage de ces dernières est égale au nombre d'éléments qu'elles contiennent (le nombre de paramètres θ_{ik}^{l} représentant $P(X_i = k \mid Pa_i = l)$) multiplié par le nombre de bits nécessaires pour stocker la valeur numérique de chacun des paramètres correspondants.

Exemple 4.3
Si un sommet qui peut prendre 5 valeurs distinctes a 4 parents pouvant chacun prendre 3 valeurs distinctes, nous auront besoin de $3^4 \times (5-1)$ *paramètres pour spécifier la table de probabilité conditionnelle.*

La longueur de la description des paramètres d'un réseau Bayésien (comprenant n CPD) est donc égale à $\sum_{i=1}^{n} d \cdot (s_i - 1) \prod_{j \in Pa_i} s_j$ où d est le nombre de bits utilisé pour stocker chaque valeur numérique, s_i est le nombre de modalités d'un sommet X_i et Pa_i est la liste de ses parents. Notons que pour un problème d'apprentissage particulier n et d sont constants.

Au final, la longueur de codage du modèle complet (structure et paramètres) est donnée par l'équation

$$\sum_{i=1}^{n} [k_i \log_2(n) + d(s_i - 1) \prod_{j \in Pa_{X_i}} s_j] \tag{4.24}$$

Il apparaît donc que les graphes fortement connectés nécessiteront une longueur de codage plus importante. Pour la plupart des sommets, la liste des parents sera plus large et la taille des tables de probabilités conditionnelles associées augmentera en conséquence. Le score MDL aura donc tendance à favoriser les réseaux dans lesquels le nombre de parents par sommet est plutôt faible.

Codage des données Lam et Bacchus ont montré que la longueur de codage des données est une fonction croissante monotone de l'*entropie croisée* (également appelée *divergence de Kullback-Leibler*) entre P, la distribution définie par la modèle candidat (un réseau Bayésien) et Q, la véritable distribution (la distribution objectif) :

$$D_{KL}(P \mid\mid Q) = \sum_{\mathbf{x}} P(\mathbf{x}) \cdot \log \frac{P(\mathbf{x})}{Q(\mathbf{x})} \tag{4.25}$$

où l'on somme sur toutes les configurations possibles $\mathbf{x} \in \mathbf{X}$ des n variables du modèle. Notons que le nombre total de configurations possibles dépend également du nombre de modalités de chacune des variables. D'après le théorème de Gibbs, la divergence de Kullback-Leibler est une quantité qui est toujours positive et qui n'est nulle que lorsque $P = Q$. Dans les faits, cette quantité est généralement impossible à calculer. Pour un réseau Bayésien ne contenant que des variables booléennes, le nombre de configurations possibles — et donc le nombre de termes à considérer — s'élève à 2^n. Bien qu'il ne soit pas envisageable de calculer une telle somme, il est possible de calculer efficacement l'entropie croisée en utilisant les marginales obtenues suite à la

factorisation de la loi jointe représentée par P en fonction de la structure S du modèle.

D'après le théorème suivant, l'entropie croisée est minimisée si et seulement si la somme des poids caractérisant chacune des CPD issues de la factorisation de P est maximisée. On définit tout d'abord la notion de poids d'un modèle.

Définition 4.11 (Poids d'un réseau Bayésien)
Le poids global d'un réseau (S, Θ) est défini comme suit :

$$\sum_{i=1|Pa_i \neq \emptyset}^{n} W(X_i, Pa_i) \tag{4.26}$$

où $W(X_i, Pa_i)$ est une mesure de poids définie pour chaque sommet en fonction de la liste de ses parents au sein du DAG :

$$W(X_i, Pa_i) = \sum_{x_i, pa_i} P(x_i, pa_i) \cdot \log_2 \frac{P(X_i, Pa_i)}{P(X_i) \cdot P(Pa_i)} \tag{4.27}$$

On somme sur toutes les valeurs possibles de la variable X_i (notées x_i) et de ses parents Pa_i (notées pa_i). Notons que le nombre de termes au sein de cette somme croît de manière exponentielle avec $\mid Pa_i \mid$. Afin de calculer efficacement ce critère on doit donc se limiter à des parentés de taille raisonnable, *a fortiori* si le nombre de modalités des variables de Pa_i est élevé.

Théorème 4.1
$D_{KL}(P \mid\mid Q)$ est une fonction monotone décroissante du poids global du modèle.

Ce théorème de Lam et Bacchus montre donc que les structures pourvues d'un poids global plus important sont plus proches, en terme d'entropie croisée, de la véritable distribution, et assurent par conséquent une longueur de codage plus faible des données.

Au final l'évaluation d'un modèle candidat est donc réalisée au moyen du score MDL, obtenu en additionnant les équations (4.24) et (4.26).

$$Score_{MDL}(S) = \sum_{i=1}^{n} [k_i \cdot \log_2(n) + d \cdot (s_i - 1) \prod_{j \in Pa_{X_i}} s_j] + \sum_{i=1|Pa_i \neq \emptyset}^{n} W(X_i, Pa_i) \tag{4.28}$$

4.3.3.2 Stratégies d'exploration de l'espace de recherche

Après avoir présenté les principaux scores permettant d'évaluer la pertinence d'une structure candidate, il reste à définir une méthode permettant de sélectionner le DAG les maximisant. Théoriquement, la recherche du graphe maximisant un certain critère de qualité est triviale : il suffit de calculer le score de chacune des structures possibles du modèle et de retenir celle dont la valeur de score est la plus élevée. Dans la pratique, une recherche exhaustive est cependant irréalisable à cause du nombre considérable de DAG pouvant couvrir un ensemble de variables

X. Pour n variables, le nombre de DAG possibles nous est donné par la formule de récurrence suivante [Rob77] :

$$r(n) = \sum_{k=1}^{n} (-1)^{k-1} \cdot \binom{n}{k} \cdot 2^{k \cdot (n-k)} \cdot r(n-k) \tag{4.29}$$

avec $r(0) = 1$. On constate que le nombre de DAG augmente de manière exponentielle avec le nombre de variables du problème : $r(1) = 1$, $r(2) = 3$, $r(3) = 25$, $r(5) = 29281$, $r(10) \simeq 4.2 \times 10^{18}$. Au-delà de 5 ou 6 sommets, il est donc impossible d'effectuer un parcours exhaustif de l'espace de recherche en un temps raisonnable. C'est pourquoi il est nécessaire de recourir à des stratégies heuristiques assurant un parcours partiel mais pertinent de l'espace de recherche afin de trouver des modèles ayant un score élevé, sans avoir à énumérer tous les DAG possibles.

Globalement, les heuristiques de recherche que nous évoquons par la suite fonctionnent toutes selon le même principe. Il s'agit de processus itératifs qui, partant d'une solution aléatoire ou préalablement choisie par l'utilisateur, proposent un parcours de l'espace de recherche visant à trouver à chaque itération une solution de qualité supérieure ou égale à celles rencontrées auparavant. À chaque itération, on dispose d'une *solution courante* S_c. Des *opérateurs de variation* sont appliqués à S_c pour générer un ensemble de *solutions candidates* $\mathbf{S_V}$ appartenant au *voisinage* de S_c. Les solutions candidates sont ensuite évaluées grâce à un score. Une *fonction de décision* permet de décider si un élément de $\mathbf{S_V}$ doit remplacer S_c, en fonction de la valeur de leurs scores. Ce processus se poursuit jusqu'à ce qu'un *critère d'arrêt* soit satisfait, par exemple lorsqu'aucune amélioration de score n'est constatée d'une itération sur l'autre.
Pour un espace de recherche donné, les heuristiques varient essentiellement en fonction des opérateurs de variation, de la définition d'un voisinage ainsi que de la fonction de décision utilisés. L'initialisation ainsi que le critère d'arrêt peuvent également être sujets à discussion. Cependant, avant d'envisager l'étude d'une heuristique de recherche, il est nécessaire de définir l'espace au sein duquel on souhaite opérer.

Définition de l'espace de recherche Définir un espace de recherche revient à choisir une représentation pour les modèles que l'on souhaite sélectionner. Ce choix est lourd de conséquences dans la mesure où il détermine la nature des opérateurs de variation mis en œuvre dans une heuristique de recherche. Ces derniers permettent de se déplacer d'un point à l'autre de l'espace de recherche en s'appuyant notamment sur une notion de voisinage qui est propre à l'espace de recherche considéré.

graphes orientés sans cycle Dans le cas le plus simple, on considère l'espace des DAG. Le voisinage d'un DAG courant S_c est l'ensemble $\mathbf{S_V}$ des DAG qui ne diffèrent de S_c que par la présence ou l'absence (et parfois l'orientation) d'un arc. Les opérateurs de variation sont donc des opérateurs de suppression, d'addition ou d'inversion d'arc. Il est important de noter que les deux derniers ne peuvent être appliqués à S_c que si le graphe résultant respecte la contrainte d'acyclicité des modèles. La génération du voisinage d'un DAG est illustrée à la figure 4.4. L'espace des DAG est un espace pertinent dont les éléments correspondent directement aux modèles que nous souhaitons évaluer et sélectionner et dont les opérateurs de variation sont simples à mettre en œuvre.

Classes d'équivalences Markoviennes Chickering [CDD96] ainsi que Auvray et Wehenkel [AW02] ont proposé de restreindre l'espace de recherche à l'espace des classes d'équivalence

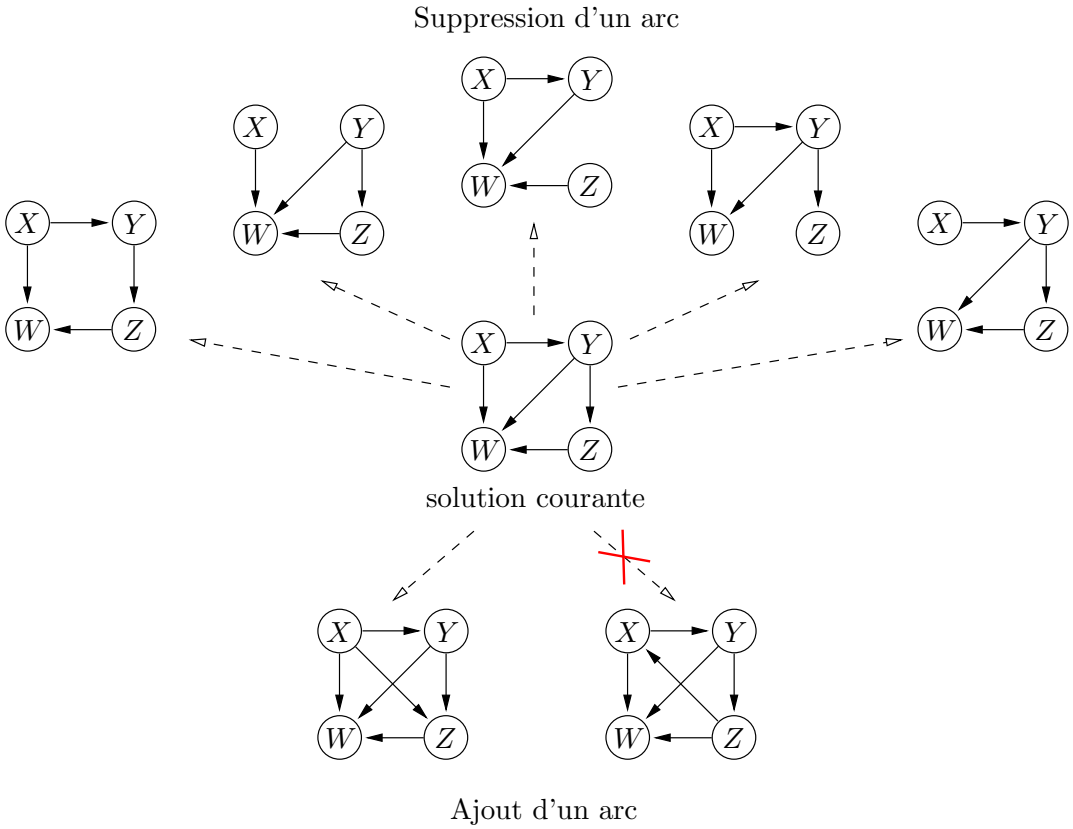


Fig. 4.4 – Construction du voisinage d'un graphe orienté sans cycle par ajout et suppression d'arcs. On constate que lorsque l'on ajoute un arc, il est possible de générer un cycle dans la nouvelle structure candidate. De tels graphes ne sont pas inclus dans le voisinage du graphe courant. Il est également possible d'étendre ce voisinage en inversant un à un les arcs du graphe courant.

Markoviennes représentées par des CPDAG (voir section 4.1.3 page 83). L'intérêt de ce choix est d'éviter l'évaluation de DAG distincts encodant les mêmes indépendances conditionnelles et ayant des valeurs de score identiques. Les auteurs avancent qu'il est possible de gagner un temps précieux en évitant d'effectuer la recherche au sein d'une classe d'équivalence dont il est impossible de distinguer les membres à partir des seules données.

Fondamentalement, l'idée de s'intéresser à l'espace des classes d'équivalence semble naturelle. Cependant, lorsque l'ordre des variables est connu ou qu'une information *a priori* permet de privilégier l'orientation de certains arcs, il devient possible de trancher entre différents DAG appartenant à la même classe d'équivalence Markovienne. Comme ces derniers cessent d'être équiprobables il devient absurde de continuer à explorer l'espace des classes d'équivalence Markoviennes. Or, il se trouve que l'utilisation de connaissances *a priori* est très utile pour apprendre des modèles de régulations complexes avec peu de données. Il est d'autant plus judicieux d'envisager leur utilisation que les biologistes disposent à l'heure actuelle de nombreuses connaissances parcellaires concernant les réseaux de régulations. Dans le même ordre d'idée, l'utilisation de données d'intervention pour lesquelles la valeur de certaines variables est fixée peut permettre d'orienter certains arcs et donc de distinguer des DAG appartenant à la même classe d'équivalence. Dans la mesure où les données d'intervention sont appelées à être de plus en plus utilisées en génomique fonctionnelle, il apparaît difficile de tourner le dos à cette source potentielle de données. Dans nos travaux (qui sont essentiellement méthodologiques) nous avons fait le choix de nous intéresser à l'exploitation de données d'observation statiques et d'écarter l'utilisation d'*a priori*. Cependant, il nous semble important de favoriser la généralité des approches d'apprentissage afin de pouvoir revenir sur ces choix si nécessaire. D'une manière générale, l'utilisation de données d'intervention ou de connaissances a priori constitue un avantage conséquent lorsque l'on souhaite traiter un problème biologique concret. Il est donc préférable de travailler dans l'espace des DAG afin de pouvoir utiliser différents types de connaissances et de données grâce auxquelles on apprend des DAG et non des classes l'équivalence Markoviennes.

Ordres topologiques Friedman et Koller [FK03] ont quant à eux proposé d'effectuer la recherche parmi les ordres topologiques des sommets du modèle étudié. Bien qu'un ordonnancement des variables ne présente aucun intérêt en soi, il existe des algorithmes permettant de trouver efficacement un DAG maximisant un score donné lorsque ses sommets sont ordonnés. C'est notamment le cas de l'algorithme K2 que nous présentons par la suite (page 105). Il s'agit donc de décomposer le problème initial en deux problèmes imbriqués : chercher l'ordre pour lequel on parvient à trouver le DAG maximisant un score. Selon Friedman et Koller, l'espace des ordonnancements des sommets est plus petit et plus régulier que l'espace des DAG, ce qui théoriquement simplifie le processus de recherche. Le même argumentaire que pour les classes d'équivalence Markoviennes nous amène à lui préférer l'espace des DAG. Ce dernier nous paraît plus générique car il implique la manipulation de graphes et d'indépendances conditionnelles que l'on retrouve dans tous les modèles graphiques.

La plupart des méthodes de recherche que nous présentons par la suite peuvent être appliquées à ces différents espaces bien que dans la majorité des cas, elles soient utilisées pour explorer directement l'espace des DAG.

Les méthodes de recherche génériques Les méthodes de recherche que nous présentons ici sont générales : elles ne concernent pas un domaine d'application en paticulier. Elles produisent une solution unique (généralement un DAG) maximisant l'un des scores que nous avons présentés

précédemment.

La recherche gloutonne L'algorithme de recherche le plus simple que l'on puisse imaginer est la montée de colline par recherche gloutonne. D'une manière générale les méthodes gloutonnes sont très répandues en optimisation. Leur fonctionnement peut être décrit de la manière suivante :

1. On choisi un graphe S_c comme point de départ de l'algorithme. Classiquement, il s'agit d'un graphe vide ou d'un graphe sélectionné aléatoirement.
2. On construit $\mathbf{S_V}$ le voisinage du graphe courant (S_c) par ajout, suppression ou inversion d'un arc à la fois, en veillant à produire systématiquement un graphe orienté sans cycle.
3. On calcule le score (typiquement, la probabilité *a posteriori*) de chacun des graphes candidats appartenant à $\mathbf{S_V}$.
4. On choisi parmi ces derniers celui dont le score est le plus élevé.
5. Si son score est supérieur à celui de la solution courante, il la remplace et on retourne à l'étape 2. Sinon, la solution courante n'ayant pu être améliorée, la procédure s'arrête.

On remarque que le calcul du score des graphes candidats est beaucoup plus simple si l'on dispose d'un score décomposable. Dans la mesure où ces graphes ne diffèrent du graphe courant que par l'ajout ou la suppression d'un arc, il est inutile de recalculer le score de S_c dans sa globalité. Pour chaque solution de $\mathbf{S_v}$, il suffit de recalculer le score local de la famille présentant un parent en plus ou en mois par rapport à S_c.

L'algorithme de la montée de colline présente l'avantage d'explorer l'espace des DAG sans la moindre restriction. Il est certes possible de formuler des hypothèses visant à limiter la taille de l'espace de recherche. Par exemple, il est courant d'imposer une borne supérieure sur le degré entrant des sommets du graphe afin de limiter sa connectivité. Toutefois de telles contraintes, bien que souhaitables dans les faits, ne sont pas nécessaires à la mise en œuvre de cet algorithme.

Le principal inconvénient des algorithmes de recherche gloutonne réside dans le fait qu'ils convergent vers l'optimum local le plus proche de la solution initiale. Dans la mesure où l'espace de recherche croît exponentiellement avec le nombre de sommets, trouver le graphe d'interaction optimal parmi les graphes dont les sommets ont au moins k parents est un problème NP-complet lorsque $k > 1$.

Il existe de nombreuses approches ayant pour objectif d'améliorer la procédure de recherche du DAG optimal en restreignant l'espace de recherche à un ensemble de solutions candidates jugées plus « judicieuses ». Ces dernières s'appuient généralement sur des hypothèses simplificatrices. Dans le cadre de la bio-informatique, elles visent également à exploiter des sources d'informations hétérogènes. Dans l'immédiat, nous allons poursuivre l'étude de méthodes génériques, les approches appliquées étant abordées dans un second temps.

Restreindre l'espace de recherche à l'espace des arbres Les travaux les plus précoces concernant l'apprentissage de structure dans les réseaux Bayésiens sont ceux de Chow et Liu [CL68] visant à apprendre des réseaux structurés sous la forme d'arbres à partir de données statiques.

La particularité de cette approche vient du fait que l'on restreint l'espace de recherche à une sous-classe des DAG : les arbres orientés. Soit T l'ensemble des arbres ayant $\mathbf{X}$ pour sommets, et $P^t(\mathbf{X})$ une distribution jointe sur $\mathbf{X}$ définie par un réseau Bayésien ayant pour structure un arbre $t \in T$. On souhaite trouver t tel que $P^t(\mathbf{X})$ fournisse l'approximation la plus fidèle possible à la distribution objectif $P(\mathbf{X})$, dont les données D sont un m échantillon. Cela revient à chercher t tel que $P^t(\mathbf{X})$ minimise la divergence de Kullback-Leibler $D_{KL}(P^t \mid\mid P)$ décrite par l'équation (4.25). Chow et Liu ont énoncé un théorème permettant de résoudre ce problème d'optimisation

sans qu'il soit nécessaire de parcourir T de manière explicite. Le théorème suivant fait appel à la notion d'information mutuelle définie par l'équation (3.27) :

Théorème 4.2 (Chow et Liu)
On attribue à tout arc $(X_i, X_j) \in \mathbf{X} \times \mathbf{X}$ un poids $W(X_i, X_j)$ égale à l'information mutuelle entre ses extrémités $I(X_i, X_j)$. Parmi tous les arbres $t \in T$, celui dont la distribution $P^t(X)$ minimise l'entropie croisée $D_{KL}(P^t \,||\, P)$ est l'arbre de recouvrement maximum.

Cette tâche de minimisation peut donc être résolue grâce à un algorithme standard de résolution du problème de l'arbre de recouvrement de poids maximum tel que l'algorithme de Kruskal ou l'algorithme de Prim. On remarquera en outre que cette méthode permet de trouver un arbre minimisant la divergence de Kullback-Leibler sans qu'il soit nécessaire de calculer cette dernière, tous les calculs portant sur des coût locaux (en l'occurrence des informations mutuelles).
On notera également que d'autres types de coût locaux peuvent être utilisés. Heckerman et col. [HGC95] ont suggéré d'utiliser un score quelconque, localement décomposable, pour définir le poids d'une arête :

$$W(X_i, X_j) = score(X_i, X_j) - score(X_i, \emptyset) \tag{4.30}$$

où $score(X_i, X_j)$ est le score local en X_i lorsque X_j est son parent, et $score(X_i, \emptyset)$ est le score local de X_i s'il n'a aucun parent.

Les algorithmes de Kruskal ou de Prim produisent un arbre non orienté dont les arcs peuvent être orientés en choisissant une racine parmi les sommets de $\mathbf{X}$ puis en parcourant l'arbre à l'aide d'un algorithme de recherche en profondeur. Cette racine peut être choisie aléatoirement ou à l'aide de connaissances *a priori*.

Si la distribution $P(\mathbf{X})$ a une structure arborescente alors, pour un nombre suffisant de données, elle peut être retrouvée avec exactitude. Dans le cas contraire, la méthode de Chow et Liu garantie que la structure apprise est, parmi tous les arbres possibles, la plus proche de la véritable distribution. Malgré ses nombreux avantages, cette approche est limitée à l'apprentissage de modèles structurés en arbres. Rebane et Pearl (1987) ont étendu la méthode de Chow et Liu à la découverte de réseaux simplement connectés (ou poly-arbres). Bien qu'elle nécessite de calculer des statistiques d'ordre supérieur, cette classe de modèles permet de décrire des interactions plus riches que les arbres. De nouveau, si $P(\mathbf{X})$ a une structure en poly-arbre, cette distribution peut être apprise avec précision sans quoi elle peut être très imprécise. Cependant cette approche ne permet toujours pas d'apprendre des DAG qui permettent de capturer avec une plus grande précision les interactions multiples des réseaux de régulation.

Imposer un ordonnancement des variables Cooper et Herskovits [CH92] ont proposé de simplifier l'exploration de l'espace des DAG en imposant une contrainte sur l'espace de recherche : chaque structure candidate S doit respecter un ordre topologique (voir définition 4.4) spécifié par l'utilisateur. Par conséquent, pour un ordre topologique donné $\pi : \{X_1, \ldots, X_n\} \mapsto \{r_1, \ldots, r_n\}$, tout sommet X_i d'un graphe candidat ne peut avoir comme parent qu'un sommet X_j de rang supérieur : $r_j > r_i$. On dit alors que X_j est un parent potentiel de X_i. Les auteurs ont ensuite proposé une procédure de recherche gloutonne tirant parti de cette contrainte sur la direction des arcs dans les solutions candidates : l'algorithme K2. Partant d'un graphe vide, l'algorithme K2 construit la parenté Pa_i de chacun des sommets $X_i \in \mathbf{X}$ du réseau de la manière

suivante :
K2 essaie d'ajouter itérativement un parent X_j à un sommet X_i tel que $r_j > r_i$. À chaque itération, il sélectionne le sommet qui, parmi tous les parents potentiels de X_i, assure l'augmentation la plus importante du score choisi une fois l'arc $X_j \rightarrow X_i$ ajouté. Ce processus est répété pour tous les sommets jusqu'à ce qu'aucun ajout ne permette d'augmenter le score.

Des approches similaires à celle de Cooper et Herskovits ont été proposées par la suite. Elles diffèrent essentiellement de l'originale par la fonction de performance exploitée. Alors que Cooper et Herskovits ont proposé de maximiser le score BD, Bouckaert [Bou93] a proposé une variante à l'algorithme K2 utilisant le score MDL. De la même manière, il est possible d'utiliser le score BIC.

L'utilisation d'un ordre topologique permet de réduire l'espace des DAG candidats mais aussi de distinguer des DAG appartenant à une même classe d'équivalence dans la mesure où π fixe l'orientation des arcs. Cooper et Herskovits ont également proposé de limiter la taille des listes de parents construites pour chaque sommet. Cette contrainte supplémentaire permet de limiter la taille de l'espace de recherche et de diminuer la complexité du calcul des scores locaux. Elle suppose que la connectivité d'un réseau de régulation est limité et qu'un gène a un nombre de régulateurs limité ce qui, dans la majorité des cas, semble pertinent.

Malgré les nombreux attraits de cette approche, elle demeure dans les faits difficilement applicable en l'état, l'ordre topologique des variables étudiés étant une connaissance difficilement accessible. L'algorithme K2 est plus particulièrement utilisé comme une fonction de décodage dans le cadre de méthodes dédiées à l'exploration de l'espace des ordres topologiques. Pour un ordre donné, K2 permet de convertir ce dernier en un DAG candidat. Le score de ce DAG sert à évaluer l'ordre topologique à partir duquel il est construit. Bien sûr, cela suppose que K2 trouve le DAG maximisant le score. Ce dernier point est discutable dans la mesure où K2 est un algorithme glouton qui ne prétend trouver qu'une optimum local de ce score. Malgré cela, l'avantage prépondérant apporté par la connaissance de l'orientation des arcs permet à K2 de trouver de très bonnes solutions lorsque l'ordre topologique proposé concorde avec celui de la distribution objectif.

Présélectionner les parents potentiels Friedman et col. [FNP99] ont développé une procédure itérative de parcours de l'espace des structures parcimonieuses (ou SCA, pour *sparse candidate algorithm*). À chaque itération, leur algorithme alterne une étape de restriction de l'espace des solutions candidates et une étape de recherche de la structure optimale au sein du sous-espace défini.

La première étape, visant à réduire la taille de l'espace de recherche, définit pour chaque sommet X_i un sous-ensemble de nœuds $C_i \in \mathbf{X} \backslash \{X_i\}$ constituant une liste de parents potentiels. Cette dernière est constituée de Pa_i, la liste des parents préalablement identifiés lors des étapes de recherche antérieures, et d'un sous-ensemble des sommets restants, maximisant une mesure de dépendance vis-à-vis de X_i. Classiquement, cette dépendance est mesurée grâce à l'information mutuelle et à l'information mutuelle conditionnelle. Afin de renforcer la contrainte sur la taille de l'espace de recherche, un paramètre k spécifié par l'utilisateur fixe également la taille maximum de la liste des parents potentiels.

La seconde étape consiste à trouver un DAG maximisant un critère de qualité tel que le score BDe [HGC95] choisi par les auteurs, tout en respectant les contraintes définies à l'étape précédente : les parents de chacun des sommets X_i d'une solution candidate devront appartenir à la liste C_i des parents potentiels qui a été proposée. Cette phase d'optimisation est assurée par un algorithme de recherche gloutonne augmenté par une stratégie *tabou*. Celle-ci consiste

simplement à entretenir une liste énumérant les N (la taille de la liste étant fixé par l'utilisateur) derniers DAG rencontrés et à interdire à l'algorithme de recherche de considérer une solution appartenant à cette liste. Il s'agit d'une méthode simple permettant d'échapper à certains optima locaux.

Ces deux étapes sont répétées jusqu'à ce qu'aucune amélioration de la solution courante ne puisse être trouvée.

Restreindre l'espace de recherche dans les réseaux de régulation génétique Les méthodes présentées jusque là peuvent être appliquées à n'importe quel type de problème modélisé par un réseau Bayésien. Il est cependant possible de développer des approches plus spécifiques à l'apprentissage de réseaux de régulation génétique.

Dans le même ordre d'idée que l'algorithme SCA présenté ci-dessus, l'algorithme du parent idéal de Nachman et col. [NRF04] sélectionne les parents des sommets d'une solution candidate en fonction de leur proximité avec un parent « idéal ». Plus précisément, les auteurs proposent de construire le profil d'expression d'un parent (régulateur) qui est idéal dans la mesure où il permet de prévoir de manière précise le profil d'expression de sa cible (gène régulé). Les parents potentiels sont alors recrutés en fonction de la similarité de leur profil avec celui de ce parent idéal et virtuel.

A contrario, au lieu d'apprendre un réseau Bayésien complet, Peña et col. [PnBT05] ont proposé de se concentrer sur la découverte du voisinage d'un gène d'intérêt au sein du réseau de régulation. Pour cela, ils choisissent un gène cible et font grandir le réseau de manière itérative à partir de ce dernier. La procédure, qui ne nécessite que des calculs de scores locaux, s'arrête après un nombre d'itérations pré-défini, chacune d'entre elle consistant à rechercher les enfants et les parents des sommets préalablement inclus dans le graphe.

Initialiser une recherche gloutonne avec un premier algorithme La plupart des algorithmes que nous avons présentés reposent sur des approches gloutonnes et sont donc particulièrement sensibles à l'initialisation. Pour accroître les chances de trouver le maximum global, il est courant de recourir à des initialisations multiples afin de générer plusieurs solutions potentielles parmi lesquelles ont choisira celle dont le score est le plus élevé. Une autre approche consiste à utiliser un algorithme peu sensible à l'initialisation mais produisant des modèles simplifiés afin de fournir une solution de départ pertinente pour une second algorithme, moins contraint, mais sensible à la solution initiale.
Leray et Francois [FL04] ont proposé d'utiliser l'arbre construit par l'algorithme MWST comme point de départ pour un algorithme de montée de colline. Ils ont également proposé d'utiliser l'arbre obtenu par MWST pour générer un ordre topologique qui est ensuite exploité par l'algorithme K2. Ces différentes approches ont entre autre été comparées avec les résultats obtenus par un algorithme K2 initialisé avec différents ordres topologiques tirés aléatoirement. Pour une présentation plus récente et complète de ces travaux, il est possible de se référer à la thèse d'Olivier François [Fra06].

Échantillonner l'espace de recherche Afin d'envisager l'apprentissage de structure dans un cadre Bayésien, au lieu d'utiliser des heuristiques de parcours déterministes, de nombreux travaux ont proposé de recourir à des techniques de simulation telles que les méthodes MCMC (pour *Markov Chain Monte Carlo*). Celles-ci permettent d'échantillonner la distribution de probabilité *a posteriori* $P(S \mid D)$ des structures d'un réseau Bayésien. Cette approche permet d'envisager l'apprentissage de structure dans un cadre Bayésien au lieu de fournir une estimation ponctuelle

de la structure comme c'est le cas avec l'approche du maximum *a posteriori*. Dans les grandes lignes, la méthode MCMC, lorsqu'elle est appliquée à l'espace des DAG, peut être décrite de la manière suivante :
Partant d'une structure le plus souvent aléatoire, un voisin est généré par ajout, suppression ou inversion d'un arc choisi au hasard. Cette nouvelle structure est acceptée ou rejetée en fonction du critère de décision de Metropolis et Hastings [Has70]. En répétant cette procédure on produit une chaîne de Markov qui, sous certaines conditions, converge en distribution vers la distribution *a posteriori*.
Cette approche a été appliquée à l'espace des DAG [LBU04], des classes d'équivalences de Markov, et des ordres topologiques [FK03]. Enfin, Husmeier a également utilisé la méthode MCMC pour apprendre des réseaux Bayésiens dynamiques [Hus03].

4.3.4 Méthodes hybrides

Divers travaux visant à mêler l'apprentissage par contraintes avec celui à base de scores ont vu le jour afin de conjuguer les avantages respectifs de ces deux approches. L'algorithme BN-PC-B [CGK+02] proposé par Cheng et col. est l'un des plus connus. Ce dernier tire parti de l'arbre construit par l'algorithme MWST de Chow et Liu pour limiter le nombre de tests d'indépendance conditionnelle nécessaires dans les deux premières étapes de ces algorithmes :

1. Un arbre non orienté est construit grâce la méthode de Chow et Liu.
2. Un nombre restreint de tests d'indépendance conditionnelle est réalisé afin d'ajouter des arêtes manquantes à cet arbre.
3. Une dernière série de tests permet d'élaguer le graphe (élimination d'arêtes surnuméraires) et de détecter les V-structures (orientation partielle du graphe).

Finalement, les arcs non orientés appartenant au graphe partiellement orienté sont orientés en suivant la même méthode que pour les algorithmes IC et PC. Une seconde version de cet algorithme appelée BN-PC-A propose de diminuer le nombre de tests d'indépendance conditionnelle réalisés par la prise en compte d'un ordre topologique des sommets permettant d'orienter les arêtes dès la première phase de l'algorithme.

A contrario, l'algorithme MMHC (pour *Max-Min Hill Climbing*) de Tsamardinos et col. [TBA06] recourt aux méthodes à base de scores dans la phase finale de la procédure d'apprentissage. Il construit le squelette d'un réseau Bayésien au moyen de tests d'indépendance conditionnelle selon le même principe que l'algorithme PC [PSS00] avant d'utiliser une recherche gloutonne afin d'orienter les arcs. Cette dernière utilise le score BD [HG95] avec un *a priori* uniforme pour évaluer les solutions candidates.

4.4 Réseaux bayésiens dynamiques

Bien que nous n'ayons pas travaillé sur cette famille de modèles, il nous semble incontournable de citer les réseaux Bayésiens dynamiques qui constituent une alternative importante lorsque l'on dispose de données cinétiques. La thèse de Kevin Murphy [Mur02] constitue un travail de référence concernant cette classe de modèles.
Lorsque les variables auxquelles on s'intéresse varient dans le temps, la famille des réseaux Bayésiens dynamiques peut être employée. Gharamani et a. appellent *réseau Bayésien dynamique* tout modèle graphique reflétant une évolution temporelle. Le plus simple des réseau Bayésien est alors une chaîne de Markov : Soit $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T$. Pour une hypothèse Markovienne d'ordre 1, le présent ne dépend que du passé proche. Dans el cas continu, le modèle se décrit à l'aide de

l'équation suivante :

$$x_{t+1} = f_\theta(x_t) + \epsilon_t \tag{4.31}$$

où f est une fonction linéaire ou non et ϵ_t est la réalisation d'un bruit gaussien.

Dans cette famille de modèle la loi jointe peut être factorisée de la manière suivante :

$$P(\mathbf{x}_1, ..., \mathbf{x}_T) = P(\mathbf{x}_1) \prod_{i=1}^{T-1} p(\mathbf{x}_{t+1}|\mathbf{x}_t) \tag{4.32}$$

Le temps indique ici la causalité, ce qui rend plus facile l'inférence. Il est aussi possible de déployer dans le temps un réseau bayésien statique, on emploie alors soit le terme de réseau de croyance Bayésien dynamique soit le terme générique de réseau bayésien dynamique. Il n'y a plus de contrainte sur l'acyclicité du graphe qui définit les indépendances conditionnelles à travers le temps. On s'intéresse à l'évolution des variables unidimensionnelles $x_i(t)$:

$$p(x_1(1), ..., x_n(T)) = \prod_{i=1}^{n} p(x_i(1)). \prod_{t=1}^{T-1} \prod_{i=1}^{n} p(x_i(t+1)|Pa_i(t)). \tag{4.33}$$

L'apprentissage d'un tel réseau ressemble à l'apprentissage d'un réseau Bayésien dynamique : il se décompose en apprentissage de la structure et apprentissage des probabilités conditionnelles.

Il est aussi possible de complexifier le modèle Markovien si on suppose la présence d'un processus caché :

$$x_{t+1} = F_\theta(x_t) + \epsilon_t^h \tag{4.34}$$

$$y_t = H(x_t) + \epsilon_t^o \tag{4.35}$$

où (x_t) est le processus caché et $y(t)$ est le processus observé. On parle alors de modèle à espace d'états.

Dans les réseaux Bayésiens dynamiques [dHIK$^+$03], la structure et les paramètres sont appris ce qui les rend pertinents pour l'inférence des réseaux de régulation génétique. Ils constituent une alternative intéressantes aux réseaux booléens probabilistes.

Les modèles à espace d'états sont très utilisée pour apprendre des réseaux de régulation en l'absence de mesures des concentrations de protéines (les variables cachées). Ils ont donné lieu à des développement au sein du laboratoire IBISC [PRM$^+$03, dBLP$^+$05, QBdB07].

Troisième partie

Apprentissage évolutionnaire des réseaux Bayésiens

Chapitre 5

Algorithmes évolutionnaires pour l'apprentissage de structure

Dans le chapitre précédent, nous avons présenté différentes approches permettant d'apprendre la structure d'un réseau Bayésien à partir de données. La plupart d'entre elles ont été appliquées à l'apprentissage de réseaux de régulation génétique à partir de données statiques (généralement discrètes) de profils d'expression. Parmi ces approches nous avons plus particulièrement mis en avant l'exploration de l'espace des structures. Quel que soit le score utilisé pour évaluer la qualité d'une structure candidate, il apparaît clairement que la principale difficulté de ces méthodes réside dans la manière d'optimiser ce score. En effet, compte tenu de la très grande taille de l'espace de recherche [Rob77], l'utilisation d'heuristiques de recherche déterministes telles que l'algorithme K2 [CH92] ou l'algorithme de montée de colline [Chi02] est nécessaire pour identifier la structure maximisant un score. Cependant, ces méthodes ne trouvent que des solutions sous-optimales et nombre d'entre elles reposent sur des hypothèses très simplificatrices telles que la restriction de l'espace de recherche à des arbres ou à des DAG respectant un ordre topologique. Dans la mesure où il s'agit d'un problème NP-difficile [CDD96], différents travaux ont proposé d'utiliser des heuristiques stochastiques telles que les méthodes MCMC [FK03, KC01b], le recuit simulé [JN06, WTX04, HGC95], ou la programmation génétique [LHY05, CM04, WLL99] pour améliorer la recherche de solutions candidates. Ces dernières sont supposées contourner certaines limites des stratégies de recherche déterministes telles qu'une forte dépendance à l'initialisation de l'algorithme ainsi que la tendance à sombrer prématurément dans des optima locaux. Par ailleurs, il s'agit de méthodes génériques susceptibles d'être utilisées dans un large éventail de situation. Le fonctionnement global de ces algorithmes est indépendant des hypothèses simplificatrices communément utilisées ou de la nature des données mises à notre disposition.

Dans nos travaux, nous avons donc choisi d'utiliser ce type d'approches pour explorer l'espace des DAG afin d'identifier celui maximisant le score BIC (détaillé par la formule (4.18)). Nous nous sommes plus particulièrement intéressé aux algorithmes évolutionnaires (AE) [Hol75, Gol89] qui sont adaptés aux problèmes d'optimisation combinatoire. Dans ce qui suit, nous allons commencer par rappeler quelques principes généraux concernant cette classe d'algorithmes en évoquant les stratégies globales que nous avons retenues. Nous nous attarderons ensuite sur les choix que nous avons faits concernant les opérateurs de variation appliqués aux DAG ainsi que sur l'utilisation de techniques de spéciation également appelées *niching*. Le chapitre suivant présentera les différents résultats numériques ayant permis de sélectionner les stratégies de recherche les plus prometteuses et de les valider face à différentes méthodes d'apprentissage alternatives cou-

ramment utilisées dans la littérature.

Ces travaux ont été présentés lors de l'édition 2007 de la Conférence francophone sur l'Apprentissage automatique (CAp) [AFdB07]. Ils ont également fait l'objet d'une publication dans LNCS [AdBF07] qui a été étendue par la suite pour le journal BMC Bioinformatics [AFGdB08].

5.1 L'algorithme évolutionnaire générique

Dans ces travaux, nous nous intéressons à l'utilisation des AE pour l'apprentissage de structure dans les réseaux Bayésiens. Il ne s'agit pas d'une étude portant sur les AE en tant que tels. Nous nous concentrons sur les stratégies retenues pour accomplir notre tâche d'apprentissage et ne prétendons pas offrir une vue d'ensemble des problèmes rencontrés avec cette famille de méthodes d'optimisation.

5.1.1 Généralités

Les *individus* soumis au processus d'évolution sont des solutions candidates au problème d'optimisation qui nous intéresse. Ils appartiennent donc tous à l'espace de recherche de notre problème d'optimisation : l'espace des DAG. Ces individus sont plus ou moins performants ou — pour reprendre l'analogie Darwinienne — *adaptés* à la résolution de notre problème d'optimisation. Dans notre cas, la fonction de performance est donc égale au critère d'information Bayésien. L'ensemble des individus traités simultanément par l'algorithme évolutionnaire est appelé *population*. Ces individus évoluent au fil d'itérations appelées *générations*, jusqu'à satisfaction d'un critère d'arrêt défini par l'utilisateur.
À l'issue de chaque génération, une nouvelle population est engendrée, résultant de l'application d'un ensemble d'opérateurs aux individus de la population précédente. Les individus manipulés par un opérateur sont définis comme étant les *parents*, alors que ceux résultant de l'application de cet opérateur sont les *enfants* (ou les descendants).

À chaque boucle générationnelle, des opérateurs sont appliqués séquentiellement à la population :

1. *Sélection* des parents destinés à se reproduire afin d'engendrer m enfants, au sein d'une population de M individus.
2. *Croisement* et mutation sont appliqués aux parents sélectionnés afin de produire m enfants.
3. *Évaluation* de la fonction d'adaptation des enfants.
4. *Sélection* parmi les M individus de la population courante et les m enfants, de M individus destinés à survivre et à constituer la population à la génération suivante.

Cela implique qu'à chaque génération, le score BIC soit calculé pour chacun des enfants produits, ce qui laisse présager des temps de calcul importants.

5.1.2 Opérateurs de sélection

À chaque génération, les individus se reproduisent, survivent ou disparaissent sous l'action de deux opérateurs de sélection :

– la sélection pour la reproduction détermine combien de fois un individu se reproduit au cours d'une génération ;

– la sélection pour le remplacement détermine quels individus vont disparaître à l'issue de chaque génération, de sorte que la taille de la population demeure constante d'une génération sur l'autre, malgré la génération de nouveaux individus par reproduction.

En toute généralité, la capacité d'un individu à être sélectionné, que ce soit pour la reproduction ou le remplacement, dépend de sa performance. L'opérateur de sélection est ainsi chargé de déterminer un nombre de sélections pour chaque individu en fonction de sa performance.

En fonction des approches algorithmiques, il se peut que l'un des deux opérateurs de sélection ne favorise pas les meilleurs individus. Il est cependant nécessaire d'assurer un biais en faveur des meilleures solutions à l'issue de chaque génération.

5.1.3 Opérateurs de variation et représentation

Les opérateurs de variation, ou encore opérateurs de recherche, permettent de générer de nouveaux individus à partir des individus de la population courante. Ce faisant, ils permettent d'explorer l'espace de recherche afin de dénicher de nouvelles solutions, meilleures que les solutions courantes.

5.1.3.1 Généralités sur les opérateurs de variation

Les opérateurs de variation sont généralement classés dans deux catégories :
– les opérateurs de mutation qui, au moyen de modifications élémentaires d'un individu, permettent d'en générer un autre plus ou moins éloigné ;
– les opérateurs de croisement qui permettent de produire un ou plusieurs enfants en combinant les caractéristiques d'au moins deux parents.

Choix d'une représentation Les contraintes relatives à l'application de ces opérateurs dépendent de la *représentation* des solutions dans l'espace de recherche. La façon de modifier ou de combiner des individus est étroitement liée à leur codage. En effet, on fait évoluer une population de solutions candidates dans un espace de représentations communément appelées chromosomes, inspiré de la dualité entre génotype-phénotype :
– le phénotype est une solution du problème dans sa représentation naturelle, en l'occurrence un graphe orienté sans cycle ;
– le génotype est constitué d'une chaîne de symboles binaires et plus généralement de symboles d'un alphabet à faible cardinalité, représentant une solution candidate.

On fait donc l'hypothèse implicite qu'une solution peut être représentée par un ensemble de symboles appelés *gènes virtuels*, qui peuvent être réunis pour former une chaîne appelée *chromosome*. L'ensemble des gènes virtuels d'un chromosome particulier est appelé le *génotype*. Les différentes formes (ou valeurs) pouvant être prises par un gène virtuel sont appelées *allèles* [1]. Le génotype contient les informations nécessaires à la construction d'un individu appelées *phénotype*. Le génotype subit l'action des opérateurs génétiques : sélection et variation, tandis que le phénotype ne sert qu'à l'évaluation de la performance d'un individu.

Finalité des opérateurs de recherche Les opérateurs de recherche permettent de créer de la nouveauté dans une population en construisant des individus « descendants », qui héritent en partie des caractéristiques de leurs *géniteurs*. Ils doivent être capables d'assurer deux fonctions importantes durant la recherche d'un optimum :

[1] Dans le cas d'un chromosome binaire, chaque gène virtuel à deux allèles : 0 et 1

- l'exploration de l'espace de recherche, afin d'en découvrir les régions intéressantes, qui ont de grandes chances de contenir les optimums globaux ;
- l'exploitation de ces régions intéressantes, de façon à y concentrer la recherche et y découvrir les optimums avec la précision requise, pour celles qui les contiennent.

Par exemple, un opérateur de variation purement aléatoire, où des solutions sont tirées au hasard indépendamment les unes des autres, aura d'excellentes qualités d'exploration, mais ne pourra pas découvrir un optimum dans un temps raisonnable. Un opérateur de recherche local de type « montée de colline » pourra découvrir efficacement un optimum dans une région de l'espace , mais il y aura un grand risque pour qu'il soit local, et la solution globale ne sera pas obtenue. Un bon algorithmes de recherche et d'exploitation devra donc réaliser un équilibre adéquat entre les capacités d'exploration et d'exploitation des opérateurs de variation qu'il utilise.

Le théorème des schémas L'influence du codage et de la recombinaison sur le bon fonctionnement d'un algorithme génétique peut être comprise à travers le concept de schéma qui constitue l'un des fondements théoriques des AE. Un schéma est une configuration de certains paramètres d'une solution du problème d'optimisation. Dans le cas d'un codage binaire, cette configuration peut être représentée par une chaîne de caractères dans l'alphabet $\{0, 1, \star\}$ où $\star$ est le symbole « indéfini ». Un chromosome correspond à un schéma s'il a les même valeurs que ce dernier pour les positions où apparaissent un 0 ou un 1, et une valeur quelconque dans les positions présentant un $\star$. L'ordre d'un schéma est le nombre de symboles définis (ne correspondant pas à un $\star$).

Exemple 5.1
Un chromosome (01001) *contient entre autre les schémas* ($0\star00\star$), ($\star100\star$) *ou* ($\star1\star\star1$). *L'ordre de ces schémas est 3, 3, et 2 respectivement. Enfin le chromosome* (11001) *correspond aux schémas* ($\star100\star$) *et* ($\star1\star\star1$) *mais pas au schéma* ($0\star00\star$) *du fait de la différence en position 1.*

Holland [Hol75] explique que le moyen le plus efficace d'explorer un espace de recherche est de favoriser la reproduction des individus les plus performants. Il suppose que la performance d'un individu est due au fait qu'il contient de bons schémas. Le but de l'algorithme génétique est donc de favoriser la propagation de ces derniers au sein de la population, par le jeu de la reproduction. Le théorème des schémas [Hol75] avance que le nombre d'apparitions d'un bon schéma dans la population augmente de manière exponentielle au fil des générations. L'autre point fondamental est que le nombre de schémas différents présentés par un individu peut être important. Holland [Hol75] a également montré que le nombre de schémas traités à chaque génération est de l'ordre de n^3 avec n la taille de la population. Cette propriété, appelée *parallélisme implicite*, est l'une des raisons des performances des algorithmes génétiques.

Le concept des briques de base Selon Goldberg [Gol89] la puissance des algorithmes génétiques réside dans leur capacité à identifier et à extraire des *briques de base* (BB) des solutions recherchées. Une BB est un schéma composé de gènes virtuels fonctionnant ensemble et permettant d'augmenter la performance des individus. On parle d'*interactions* entre gènes virtuels pour désigner le fait que la capacité de chacun à améliorer la performance d'un individu dépend des autres. C'est ce qui explique que ces derniers ne puissent être optimisés de manière indépendante.

L'hypothèse sous-jacente au concept des BB est que le problème d'optimisation est, dans une certaine mesure, décomposable. On suppose qu'il est possible d'identifier des solutions partielles au problème, c'est-à-dire des sous-ensembles de paramètres pouvant être optimisés, dans un premier temps, indépendamment des autres. Le but est alors de trouver les BB correspondant

à des solutions partielles optimales et de les associer, par le jeu des croisements entre individus, afin de constituer des solutions complètes optimales. Un bon codage doit donc encourager la formation de BB en veillant à ce que les gènes qui contribuent de manière jointe à la maximisation de la fonction de performance soient proches les uns des autres au sein du génome. Cela doit permettre de les identifier et de les réassocier plus efficacement, le risque lors d'un croisement étant de « casser » des BB. Cela arrive lorsque seule une fraction des gènes virtuels composant une BB est échangée entre deux individus parents. Dans les problèmes d'optimisation complexes, il arrive cependant que les interactions entre gènes virtuels soient à la fois trop nombreuses et trop importantes pour qu'il soit possible de ranger les gènes virtuels correspondant côte à côte au sein d'un chromosome.

5.1.3.2 Mutation

L'opérateur de mutation modifie aléatoirement un individu pour en former un autre qui le remplacera. La plupart des mutations modifient un individu de telle façon que le résultat de la transformation lui soit proche. De cette façon, l'opérateur assure une recherche locale aléatoire autour de chaque individu. Dans cet ordre d'idée, la mutation peut améliorer considérablement la qualité des solutions découvertes. En effet, le croisement perd de son importance lorsqu'une grande partie de la population est localisée dans les voisinages des maxima de la fonction de performance. Dans ce cas, les individus situés sur un même maximum sont souvent identiques par le jeu de la reproduction et ne subissent aucune modification, ou alors, s'ils appartiennent à des maxima différents, les descendants montreront généralement de faibles performances. En revanche, la recherche aléatoire locale due aux mutations donne une chance à chaque individu de s'approcher des positions exactes des maxima.
La mutation avec un taux suffisamment élevé participe au maintien de la diversité utile à une bonne exploration de l'espace de recherche. Cet opérateur peut combattre les effets négatifs d'une forte pression de sélection ou d'une forte dérive génétique, phénomènes qui tendent à réduire la variance de la distribution des individus dans l'espace de recherche.
Si le taux de mutation est trop élevé et que la mutation est si forte que l'individu produit est quasiment indépendant de celui qui l'a engendré, l'évolution des individus de la population équivaut à une marche aléatoire dans l'espace de recherche et l'algorithme évolutionnaire mettra un temps excessif pour converger.

Classiquement, l'opérateur de mutation sur des chaînes binaires modifie aléatoirement les symboles d'un génotype avec une faible probabilité, typiquement de 0,1 à 0,001 par individu, égale au taux de mutation. Il existe plusieurs variétés de mutations. L'une des plus classiques est la mutation *bit-flip* avec laquelle chaque bit peut être inversé (un 1 devient un 0 et *vice versa*) indépendamment des autres avec une faible probabilité. Si le taux de mutation est trop élevé avec un grand nombre de bits mutés par individu, l'évolution des individus de la population équivaut à une marche au hasard dans l'espace de recherche et l'algorithme génétique perd son efficacité.

5.1.3.3 Croisement

Bien que nous ne soyons pas soumis aux contraintes biologiques limitant à deux le nombre d'individus participant à un croisement, nous avons choisi des opérateurs de croisement utilisant deux parents pour former deux descendants. Un opérateur de croisement est généralement stochastique dans la mesure où le croisement répété d'un même couple de parents distincts donnera

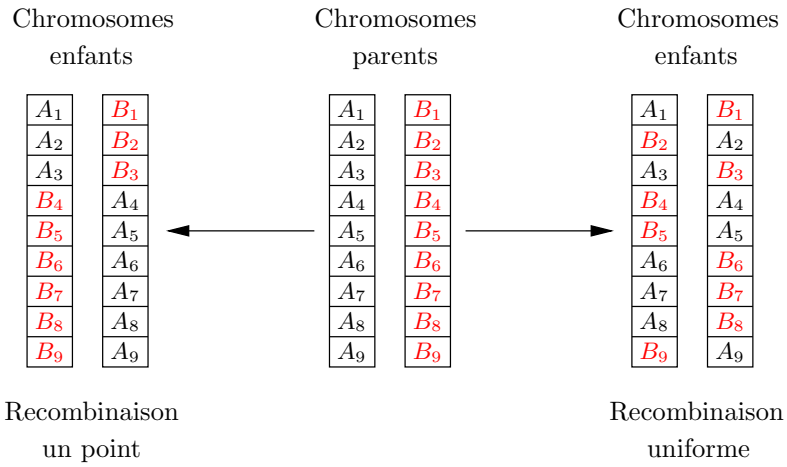


Fig. 5.1 – Exemple de recombinaison uniforme et un point.

des descendants différents. Il respecte généralement les propriétés suivantes :

- le croisement de deux parents identiques donnera des descendants identiques aux parents ;
- par extension, un indice de proximité dépendant de la représentation choisie étant défini dans l'espace de recherche, deux parents proches l'un de l'autre dans l'espace de recherche engendreront des descendants qui leur seront proches.

Pour une représentation binaire ou discrète, il existe trois variantes de croisement classiques respectant les propriétés énoncées ci-dessus :

- le croisement *un point* ;
- le croisement *deux points* ;
- le croisement uniforme.

Les croisements un point et deux points Après avoir sélectionné un couple d'individus au sein de la population, le croisement *un point* se déroule en deux étapes :

- choix aléatoire d'un point de coupure identique sur les deux chaînes binaires ;
- coupure des deux chaînes et échange des deux fragments situés au-dessous.

Ce processus produit deux descendants à partir de deux parents. Le croisement un point est le plus simple et le plus classique pour des codages utilisant un alphabet à faible cardinalité comme le codage binaire. Une généralisation immédiate de cet opérateur consiste à multiplier les points de coupure sur chaque chaîne. Pour C points de coupure, on produit ainsi $C + 1$ sous-chaînes, la moitié d'entre elles étant échangées entre les deux parents. Par exemple, pour un croisement 4 points, on produit 5 fragments de chromosome et on échange la deuxième et la quatrième sous-chaîne.

En pratique, les croisements un point et deux points sont couramment employés pour leur simplicité et leur bonne efficacité. Selon de Jong [DJ75], le croisement deux points constitue une amélioration notable du croisement un point. Dans le même temps, il avance que le fait d'augmenter le nombre de points de cassure diminue les performances de l'algorithme. Une explication à ce phénomène est que l'augmentation du nombre de points de coupure augmente la probabilité de casser des briques de base. Pour leur part, les tests que nous avons effectués n'ont montré aucune différence significative entre les croisements un point et deux points. C'est pourquoi par la suite, nous n'évoquons que le croisement un point, le croisement deux points nous étant apparu comme redondant.

Le croisement uniforme Le croisement uniforme peut être vu comme un croisement multi-points dont le nombre de coupures est indéterminé *a priori*. Pratiquement, on utilise un masque de croisement, qui est un mot binaire de même longueur que les individus. Un 0 à la n^{e} position du masque laisse inchangés les symboles à la n^{e} position des deux chaînes. Un 1 déclenche un échange des symboles correspondants. Le masque est engendré aléatoirement pour chaque couple d'individus. Les valeurs 0 ou 1 des éléments du masque sont généralement tirées avec une probabilité 0,5.

Il est difficile d'argumenter en faveur de l'une ou de l'autre des méthodes de recombinaison que nous venons de présenter [ECS89]. Selon Syswerda [Sys89], la recombinaison uniforme est plus efficace, notamment parce qu'elle est moins dépendante que les recombinaisons un ou deux points de la structure des chromosomes. En effet, les performances de ces deux méthodes chutent considérablement lorsque les recommandations que nous avons énoncées concernant les BB ne sont pas respectées [BBM93]. À l'inverse, la recombinaison uniforme continue à bien se comporter. Bien qu'ils estiment que les recombinaisons un et deux points sont optimales, Spears et de Jong [SD91] remarquent que ces dernières ne parviennent plus à générer de nouvelles solutions candidates — et donc à échantillonner de nouveaux points dans l'espace de recherche — lorsque l'algorithme converge. La recombinaison uniforme semble plus à même de produire des nouvelles solutions à partir de parents similaires. Toujours selon Spears et de Jong [JS91], la recombinaison deux points est la plus efficace pour de grandes populations mais la recombinaison uniforme est plus indiquée quand la taille de la population est faible par rapport à la complexité du problème.

5.2 Un algorithme évolutionnaire pour les réseaux Bayésiens

Nous présentons à présent les différents éléments que nous avons souhaité utiliser pour élaborer notre algorithme évolutionnaire. Nous présentons les méthodes de sélection retenues ainsi que les réprésentations et les méthodes de recombinaison que nous testons dans le chapitre suivant. Enfin, nous introduisons les méthodes de spéciation qui nous semblent être un point fondamental dans la construction d'un algorithme évolutionnaire. Pour finir, nous récapitulons nos choix et nous présentons une vue d'ensemble de l'algorithme que nous allons utiliser.

5.2.1 Opérateurs de sélection

Dans ces travaux, les individus participant à la reproduction sont choisis aléatoirement. Cela permet de laisser libre cours à l'algorithme pour recombiner des individus très différents, qui peuvent avoir des écarts de performance importants, afin de produire une grande variété de solutions candidates. Il est cependant nécessaire d'assurer un biais en faveur des meilleures solutions à l'issue de chaque génération. Nous appliquons donc la sélection lors de la phase de remplacement. Pour cela, nous avons considéré une stratégie de remplacement stationnaire, elle-même fondée sur une approche élitiste.

5.2.1.1 Remplacement stationnaire

À chaque génération, un certain nombre de descendants sont engendrés. Ils remplacent un nombre inférieur ou égal de parents, pour former la population à la génération suivante. Historiquement, la plupart des travaux ont considéré un remplacement « générationnel » consistant à remplacer l'ensemble des individus de la population par leurs descendants. Par la suite une approche diamétralement opposée s'est développée : à chaque génération, seulement un petit

nombre (typiquement 2) d'individus sont remplacés. Cette approche, appelée *remplacement stationnaire*, donne en règle générale de meilleurs résultats. Les algorithmes utilisant cette approche sont généralement appelés « algorithmes à états stationnaires ».

Le remplacement stationnaire engendre une population où les individus connaissent de grandes variations de durée de vie en nombre de générations et donc en nombre de descendants. La variance élevée de ces grandeurs favorise la dérive génétique [2], qui se manifeste d'autant plus que la population est petite.

L'utilisation d'une stratégie de remplacement stationnaire implique qu'il faille sélectionner les individus qui seront remplacés à l'issue de la reproduction. Ce choix peut être aléatoire ou dépendre de la performance des individus considérés. Pour notre part, nous avons choisi d'utiliser une stratégie de sélection élitiste.

5.2.1.2 Élitisme

Une stratégie élitiste consiste à conserver dans la population, d'une génération à l'autre, au moins l'individu ayant la meilleur performance. La performance du meilleur individu de la population courante est ainsi monotone croissante de génération en génération. Il apparaît qu'une telle stratégie augmente le taux de convergences prématurées. Elle favorise l'exploitation des meilleures solutions, se traduisant par une recherche locale accentuée, au détriment de l'exploration de l'espace de recherche.

Nous avons choisi de tirer parti des avantages respectifs de ces deux méthodes. Deux enfants sont produits à chaque génération, cependant ils entrent en compétition avec l'ensemble des individus de la génération précédente dont ils remplacent les deux individus les moins performants, s'ils ont eux-mêmes des scores plus élevés. Ainsi, le meilleur individu à la génération g est maintenu à la génération $g + 1$. De même, les nouvelles solutions candidates sont conservées, à moins qu'elles n'apportent aucune amélioration par rapport à l'un des individus de la génération précédente. La pression de sélection propre à l'élitisme permet de contrebalancer la dérive génétique caractéristique du remplacement stationnaire.

5.2.2 Représentation et recombinaison des structures de réseaux Bayésiens

5.2.2.1 Le choix de l'espace de recherche

Comme pour toute méthode de recherche, la définition d'un opérateur de variation dans les AE nécessite de définir un espace de recherche. Faire évoluer des réseaux Bayésiens est une tâche ardue et la question du codage s'avère cruciale. À cet égard, deux stratégies globales peuvent être envisagées : la recherche directe et la recherche indirecte. La recherche indirecte s'effectue généralement dans l'espace des permutations des n variables aléatoires citeHsu2002,Larranaga1996. Chaque permutation est interprétée comme un ordre topologique sur les variables. Chaque ordonnancement candidat est généralement soumis à un algorithme K2 [CH92] qui essaie de reconstruire le meilleur graphe orienté sans cycle correspondant à un ordre sur ces variables ainsi qu'aux données disponibles. Cependant, même si l'espace des permutations des n variables aléatoires est plus petit et plus lisse que l'espace des structures de réseaux Bayésiens [FK03], son exploration demeure une tâche difficile. De plus, l'algorithme K2, qui est fondé sur une méthode gloutonne, ne retrouve pas nécessairement la meilleure structure

[2] La dérive génétique est le processus par lequel certains caractères sont fixés au sein de la population par le seul fait du hasard. Un caractère est dit » fixé « lorsqu'il ne peut plus évoluer au fil des générations, typiquement lorsqu'il est partagé par l'ensemble des individus de la population.

correspondant à un ordonnancement donné des variables. L'exploration de l'espace des classes d'équivalence Markoviennes a également été entreprise par Muruzabal et Cotta [MC04].

Toutefois, l'essentiel des travaux a porté sur l'exploration directe de l'espace des graphes orientés sans cycle [LnKMY96, LnPY$^+$96, ELP97, MLD99, CT01, DBC06]. Dans ce cadre, il faut faire face au problème classique de la production de solutions incorrectes (des digraphes comportant des cycles). Cela peut être évité en imposant un ordre topologique sur les variables du modèle, comme l'a proposé Larrañaga [LnPY$^+$96], mais une telle information est généralement indisponible.

Une première réponse est apportée par Cotta et Troya [CT01] puis développée par Cotta et Muruzabal [CM02] qui proposent des opérateurs de recombinaison spécifiquement conçus pour le croisement de DAG. Bien qu'ils aient proposé plusieurs versions de leurs opérateurs, il est possible de les décrire de la manière suivante : les gènes virtuels codant les arcs des deux graphes parents sont réunis dans un unique ensemble et injectés à tour de rôle dans les futurs descendants selon un jeu de règles spécifiques visant à maintenir l'acyclicité des graphes en cours de construction. Dans une variante phénotypique de ces opérateurs, l'ordre d'inclusion des arcs dans le graphe-enfant dépend également d'une mesure d'information mutuelle entre les deux extrémités de chaque arc. Bien que ces méthodes génèrent des structures de réseaux Bayésiens correctes, elles ne permettent pas de considérer des sous-structures pertinentes dans le génome des graphes parents et de les transmettre à leur descendants. Une approche alternative est proposée par Myers et collègues [MLD99] où la production de digraphes cycliques est autorisée, les solutions incorrectes se voyant attribuées un score arbitrairement faible afin d'éviter de perdre des sous-structures potentiellement bonnes.

Dans notre étude, nous avons choisi de nous concentrer sur l'exploration directe de l'espace des DAG. Nous avons étudié des méthodes de recombinaison deux à deux génériques pour réaliser cette recherche : les croisements un point et uniforme. L'acyclicité des solutions candidates est prise en compte par un processus de réparation à l'issue de la recombinaison.

Contraintes sur l'espace des solutions et fonction de réparation La contrainte d'acyclicité est assurée *a posteriori*, en utilisant une fonction de réparation pour éliminer les cycles des nouvelles structures candidates. La fonction de réparation doit également faire respecter une seconde contrainte aux nouvelles solutions candidates. Nous souhaitons que les sommets des DAG de la population aient un degré entrant maximum fixé à 10. Les raisons de cette contrainte supplémentaire sont multiples. Elles sont avant tout d'ordre pratique et calculatoire. En limitant la taille des parentés dans les graphes testés, nous limitons la taille des tables de probabilités conditionnelles à gérer. Lorsque ces dernières sont trop volumineuses, l'estimation des paramètres nécessaire au calcul du score BIC s'avère rédhibitoire en termes de temps de calcul. Un autre avantage, en termes d'apprentissage cette fois, est qu'en agissant de la sorte, nous restreignons la taille de l'espace de recherche. Nous simplifions donc quelque peu le problème, tout en nous fondant sur l'hypothèse raisonnable que la plupart des sommets d'un réseau de régulation sont peu connectés. Enfin, cette contrainte nous permet de limiter la fréquence d'apparition des cycles, puisque ces derniers ont d'autant moins de chance d'être observés que le graphe est parcimonieux. Nous allons à présent décrire le principe de fonctionnement de la réparation.

Quels que soient le type de codage utilisé pour représenter les solutions candidates et la méthode choisie pour les manipuler, la recombinaison se résume toujours à deux DAG échangeant un sous-ensemble de leurs interactions élémentaires (les arcs). Après recombinaison, les deux graphes parents présentent de nombreuses modifications de leur topologie, qui peuvent être exprimées en termes d'additions et de suppressions d'arcs. Les suppressions d'arcs sont alors systématiquement acceptées et appliquées pour faire de la place au sein du graphe pour d'éventuelles

additions futures. En revanche, l'addition d'un arc est rejetée lorsqu'elle viole la contrainte d'acyclicité ou la contrainte sur le degré entrant des nœuds. Si tel est le cas, on essaie alors d'insérer l'arc de direction opposée, en considérant de nouveau les deux contraintes précédemment évoquées. Si l'arc inversé viole l'une d'entre elles, il est définitivement éliminé et on considère l'addition suivante. Les différents arcs à additionner sont pris en compte et testés les uns après les autres dans un ordre aléatoire.
Nous avons conscience que l'inversion d'un arc lors de son passage d'un graphe parent vers l'autre biaise significativement le processus de recombinaison. En effet, inverser un arc qui appartient à un cycle provoque l'apparition d'une V-structure, or nous avons vu que ces dernières ont une signification forte dans les réseaux Bayésiens. Cependant, nous estimons que la suppression pure et simple des arcs violant l'une ou l'autre des contraintes perturbe nettement plus le processus d'optimisation. Cette approche, qui est couramment employée, provoque une diminution importante de la connectivité au sein des réseaux Bayésiens candidats sans raison réelle.
Il est également important de noter que ce processus de réparation présente l'avantage de nous décharger dans une large mesure du problème de recherche dans une classe d'équivalence. En effet, comme nous l'avons expliqué dans le chapitre précédent, le principale avantage des algorithmes dédiés à l'exploration de l'espace des classes d'équivalence est d'éviter de se déplacer au sein d'une même classe d'équivalence. On évite ainsi de considérer de nouveaux DAG candidats encodant pourtant le même modèle statistique que les solutions courantes. Si l'on peut légitimement douter que la recombinaison de deux individus produise des enfants Markov-équivalents (entre eux ou avec leurs parents), la recombinaison rend une telle hypothèse encore plus improbable. Lorsqu'un arc est inversé pour éviter l'apparition d'un cycle, une V-structure qui n'était pas présente dans le graphe d'origine de cet arc apparaît. L'enfant produit appartient donc à une classe d'équivalence différente du graphe parent. Il en va de même lorsque la réparation se solde par une suppression d'arc. Les squelettes du graphe enfant et du graphe parent dont provient cet arc étant différents, ces deux graphes n'appartiennent pas à la même classe d'équivalence. Dans la mesure où les graphes issus de la recombinaison présentent quasi-systématiquement des cycles et doivent être réparés, il est peu probable que la reproduction induise un déplacement au sein d'une même classe d'équivalence.

5.2.2.2 Codage et recombinaison des graphes orientés sans cycle

L'efficacité de la recombinaison dépend de sa capacité à manipuler des unités d'information pertinentes. Nous avons considéré trois codages différents des DAG fondés sur différents types de gènes virtuels. À chaque fois, nous avons appliqué à ces codages l'une des méthodes de recombinaison que nous avons décrites plus haut : la recombinaison un point ou la recombinaison uniforme. La figure 5.2 résume les différents codages proposés.

Recombinaison classique Tout d'abord, nous avons opté pour une représentation classique des DAG précédemment utilisée dans la littérature. Leur structure est encodée par un vecteur de $n^2 - n$ gènes virtuels binaires notés Ψ_{ij} avec $i \neq j$, tel que $\Psi_{ij} = 1$ s'il y a un arc de X_i à X_j et 0 sinon. Ce codage dérive de la matrice d'adjacence des DAG dont les éléments sont pris un à un, colonne par colonne, afin de construire une chaîne binaire. Notons cependant que les éléments correspondant à la diagonale de cette matrice ne sont pas pris en compte dans le chromosome. Dans la mesure où ils sont toujours à 0 (un sommet ne peut pointer sur lui même) ils ne sont pas informatifs.

Ce codage peut s'expliquer à travers le concept des *briques de base* exposé dans la sous-section 5.1.3. Comme nous l'avons expliqué dans le chapitre précédent, le score BIC est décomposable et peut s'exprimer sous la forme d'une somme de scores locaux caractérisant chacune des familles

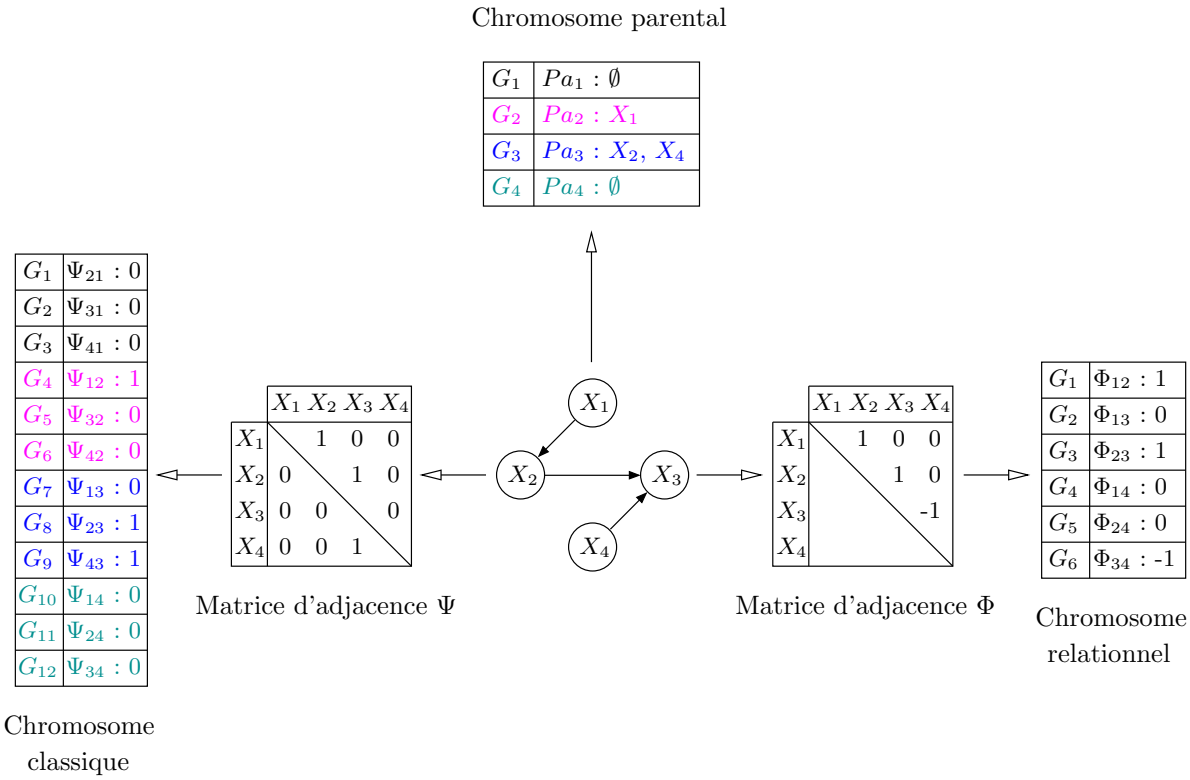


Fig. 5.2 – Les codages utilisés pour représenter des graphes orientés sans cycle en chromosomes. La construction des chromosomes relationnels et classiques est illustrée par le codage du graphe candidat par la matrice d'adjacence Ψ et la matrice triangulaire supérieure Φ respectivement. Un code couleur permet de visualiser la correspondance entre chaque liste de parents au sein du chromosome parental et la répartition des arcs correspondant à ces parents dans le chromosome classique. On remarque que les arcs définissant une liste de parents pour un sommet sont adjacents dans le chromosome classique.

du DAG. En première approche, on peut donc envisager de maximiser le score BIC en maximisant chacun de ses termes. Comme le score local d'un sommet ne dépend que de la liste de ses parents, maximiser les scores locaux implique de trouver la parenté optimale pour chacun des sommets du graphe. Bien sûr, il ne s'agit là que d'une approximation. La contrainte d'acyclité crée une dépendance entre les différentes parentés que nous mettons ici de côté en supposant que le processus d'évolution parviendra à les associer correctement. De même, les parentés ne sont pas nécessairement des sous-solutions élémentaires du problème d'optimisation. En réalité, il est possible que les parents d'un sommet puissent être pris en compte séparément, c'est-à-dire que ces derniers peuvent optimiser le score indépendamment les uns des autres. Pour faire le parallèle avec les réseaux de régulation, plusieurs régulateurs peuvent fort bien réguler un même gène de manière indépendante. Toutefois, en l'absence de connaissances *a priori* ou de données supplémentaires, il est impossible de prédire quels sont exactement les gènes virtuels qui doivent être pris en compte simultanément.

En s'appuyant sur l'hypothèse selon laquelle les parentés contiennent des solutions partielles du problème, on remarque que le chromosome classique respecte l'un des impératifs des briques de base : les gènes virtuels susceptibles de maximiser de concert le score BIC sont situés côte à côte au sein du génome. La figure 5.2 illustre cette propriété en représentant tous les gènes virtuels définissant une parenté donnée avec la même couleur. On constate que ces derniers sont répartis au sein de sous-chaînes.

Pour exploiter la répartition des gènes virtuels au sein des chromosomes classiques, nous leur appliquons un croisement un point : les chromosomes parents subissent une cassure aléatoire et s'échangent entre eux l'une des deux sous-chaînes résultantes. Ainsi, la recombinaison doit permettre d'échanger des sous-chaînes contenant potentiellement des solutions partielles. À chaque recombinaison cependant, le point de coupure apparaît dans l'une de ces sous-chaînes et la casse.

Par la suite, nous appellerons *recombinaison classique* la recombinaison un point appliquée aux chromosomes classiques.

Recombinaison parentale La seconde représentation que nous avons étudiée pour nos graphes candidats repose sur les chromosomes parentaux précédemment utilisés par Myers et collègues [MLD99]. Un chromosome parental est composé d'une séquence de n gènes virtuels, chacun d'entre eux correspondant à une liste d'adjacence Pa_j (avec $j \in \{1, \ldots, n\}$) spécifiant la liste des parents d'un sommet X_j.

Ce codage présente l'intérêt de protéger les listes parentales d'une éventuelle cassure en représentant chacune d'entre elles au sein d'un unique gène virtuel. Il est ainsi possible d'échanger les listes de parentés entre les DAG candidats afin de trouver la meilleure association possible. D'un point de vue biologique, on peut considérer que Pa_j représente un ensemble de gènes régulant l'activité transcriptionnelle du gène X_j, ce qui justifie l'échange de ces listes parentales comme unités d'information cohérentes. Dans la mesure où toutes les listes parentales sont échangées dans leur globalité, elles demeurent inchangées à l'issue du processus de recombinaison. Par ailleurs, il est peu probable que les schémas correspondant aux parentés optimales soient générés dès l'initialisation. Cela est d'autant moins probable que chaque gène virtuel peut prendre un très grand nombre de valeurs dépendant des multiples listes parentales envisageables pour chaque nœud. L'opérateur de mutation semble donc nécessaire à leur modification afin d'explorer de nouvelles configurations prometteuses.

Les chromosomes parentaux sont recombinés au moyen du croisement uniforme : deux individus s'échangent un sous-ensemble de leurs gènes virtuels présélectionnés aléatoirement. Il est inutile d'utiliser la recombinaison un point ici car l'ordonnancement des gènes virtuels au

sein des chromosomes n'a aucune signification. Par contre, la recombinaison uniforme permet de favoriser un brassage important des listes parentales entre les individus. Un taux d'échange paramètre la proportion de gènes virtuels devant être passés d'une solution parente à l'autre durant la recombinaison. Dans la mesure où ce processus est symétrique, le taux d'échange est compris entre 0 et 0,5.

Dans la suite de ce mémoire, nous appellerons *recombinaison parentale* la recombinaison uniforme lorsqu'elle est appliquée aux chromosomes parentaux.

Recombinaison relationnelle Nous avons également considéré des chromosomes relationnels codant explicitement les relations élémentaires entre variables et permettant leur échange entre graphes parents. L'idée consiste à recoder la matrice d'adjacence sous la forme d'une matrice triangulaire supérieure Φ telle que pour tout couple de sommets (i, j) tel que $i > j$:

- $\Phi_{ij} = 0$ s'il n'y a pas d'arc entre X_i et X_j ;
- $\Phi_{ij} = 1$ si $X_i \rightarrow X_j$;
- $\Phi_{ij} = -1$ si $X_j \rightarrow X_i$.

Les éléments Φ_{ij} tels que $i \leq j$ ne sont pas pris en compte. En prenant les éléments de cette matrice triangulaire supérieure un à un, colonne par colonne, il est possible de construire un vecteur ternaire correspondant à un *chromosome relationnel*.
Nous avons utilisé cette représentation compacte afin de nous débarrasser des informations redondantes présentes dans le premier codage. En effet, coder séparément les arcs orientés $X_i \rightarrow X_j$ et $X_j \rightarrow X_i$ avec Ψ_{ij} et Ψ_{ji} respectivement, apparaît inutile puisque si l'un prend la valeur 1, l'autre devra prendre la valeur 0 du fait de la contrainte d'acyclicité. On peut donc diminuer substantiellement le nombre de variables à représenter tout en étant assurer de ne jamais produire un graphe dont les sommets bouclent directement sur eux-mêmes.

La proximité des gènes virtuels le long d'un chromosome relationnel n'est pas corrélée avec leur dépendance fonctionnelle. Ici, la structure des chromosomes n'est due qu'au codage et n'a aucune signification. Par conséquent, tout comme les chromosomes parentaux, les chromosomes relationnels sont croisés grâce à la recombinaison uniforme.
Outre le fait que cette approche favorise un brassage important des caractéristiques structurales des individus de la population, elle permet également de gagner en généralité. En effet, ici, aucune hypothèse concernant la structure du problème n'est faite. Théoriquement, les gènes virtuels qui contribuent conjointement à la maximisation de la fonction de performance doivent être proches au sein du génome cependant, une telle information n'est généralement pas disponible préalablement à l'apprentissage. La méthode de recombinaison que nous proposons ici permet de tenir compte de cette réalité. Elle permet d'éviter de se retrouver piégé par un problème dans lequel les interactions entre variables sont trop nombreuses ou trop fortes pour être traitées dans le cadre conceptuel des briques de base.

Par la suite de ce mémoire, nous appellerons *recombinaison relationnelle* la recombinaison uniforme lorsqu'elle est appliquée aux chromosomes relationnels.

5.3 Préserver la diversité de la population des structures candidates

Une caractéristique fondamentale des algorithmes génétiques est leur capacité à explorer l'espace de recherche à partir de multiples points en parallèle. Cependant, alors que l'algorithme progresse, les chromosomes de la population tendent à se ressembler. Cela s'explique par le fait que les gènes virtuels de certaines solutions qui ont une performance supérieure à la moyenne

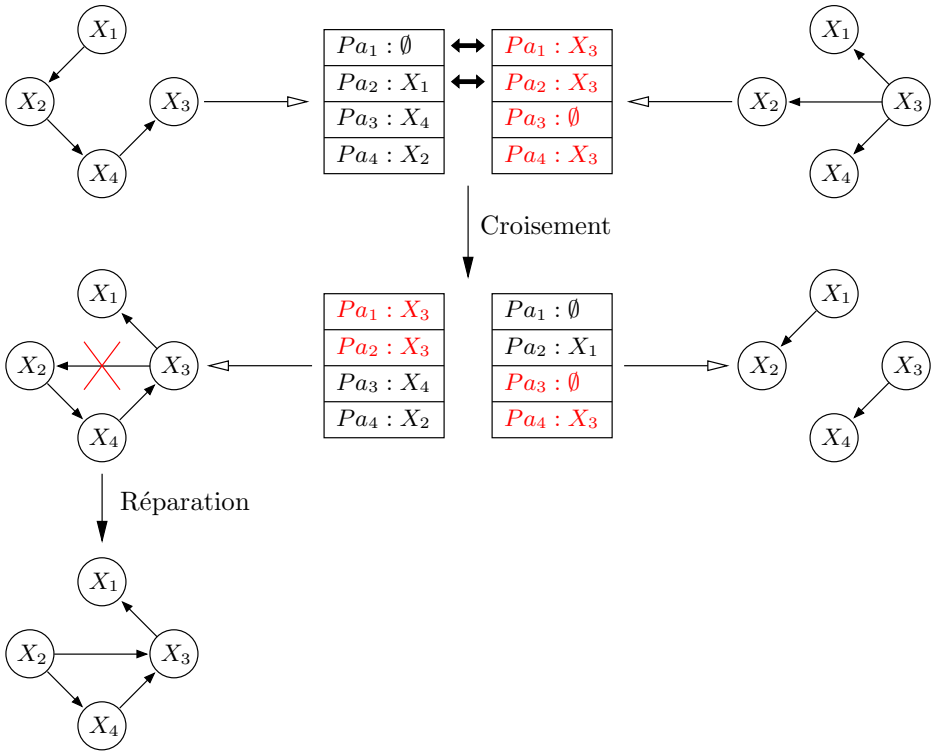


Fig. 5.3 – Exemple de recombinaison uniforme sur des chromosomes parentaux.

des individus « colonisent » la population. Ce phénomène de convergence prématurée aboutit à la formation d'une population homogène d'individus, certes performants, mais sous-optimaux. L'homogénéisation de la population est susceptible d'empêcher l'opérateur de recombinaison d'explorer de nouvelles régions de l'espace des solutions. En effet, la recombinaison de solutions identiques ou très proches produit généralement des solutions similaires aux solutions recombinées. Il existe divers mécanismes permettant de maintenir la diversité de la population afin de favoriser la génération de solutions originales et de réaliser une exploration plus profonde de l'espace de recherche. L'approche la plus simple consiste à recourir à la mutation pour introduire de la diversité dans la population. La seconde approche repose sur les méthodes de *spéciation*. Ces dernières sont supposées maintenir la diversité des solutions en limitant le champ des processus de sélection à des sous-ensembles de solutions similaires. Elles ont déjà été utilisées avec succès afin d'améliorer l'apprentissage de structures de réseaux Bayésiens avec de la programmation évolutionnaire [LHY05].

5.3.1 Introduire de la diversité par mutation

La première approche pour éviter que la population ne devienne trop homogène consiste à utiliser un opérateur de mutation. Les modifications aléatoires réalisées au sein des modèles candidats introduisent de la diversité dans la population. La mutation permet ainsi d'échapper à des minima locaux et d'explorer de nouvelles régions de l'espace de recherche.
Dans notre cas, il nous a semblé pertinent d'appliquer l'opérateur de mutation directement au phénotype des individus, c'est-à-dire à la structure même des solutions candidates. On constate en effet qu'une modification des valeurs des gènes virtuels pour les différents types de chromo-

somes considérés aboutit à des modifications topologiques élémentaires (ajout, suppression ou inversion d'arcs) des graphes qu'ils représentent. Nous avons décidé de ne pas pratiquer d'inversion d'arcs car comme nous l'avons vu, la réparation en réalise déjà un certain nombre. La mutation se caractérise donc par l'addition ou la suppression aléatoire d'arcs au sein des modèles candidats. Elle repose sur une procédure de type *bit-flip* adaptée à la manipulation de graphes orientés sans cycle.

Définition 5.1
On considère les relations entre tous les couples de variables $(X_i, X_j) \in \mathbf{X}^2$ *de telle sorte que chacune d'entre elles est susceptible d'être modifiée indépendamment des autres avec une probabilité égale au taux de mutation. Ces modifications sont telles que :*

- *s'il existe un arc entre* X_i *et* X_j*, il est supprimé ;*
- *si* X_i *et* X_j *ne sont pas connectés, un arc est ajouté. L'orientation de cet arc est choisie de manière à respecter l'acyclicité du graphe. Si les deux orientations sont possibles, l'une des deux est choisie au hasard.*

Si une borne supérieure k *sur le degré entrant des nœuds est spécifiée, l'ajout d'un arc supplémentaire ne sera autorisé que si le degré entrant du nœud cible est inférieur à* k*. Dans l'éventualité où l'ajout d'un arc viole systématiquement l'une de ces contraintes, la modification n'est pas appliquée.*

Classiquement, un taux de mutation détermine la proportion des individus mutés dans la population. Nous avons cependant fait le choix de paramétrer non pas le nombre d'individus mutés, mais le nombre de mutations survenant au sein d'un même individu. Nous sommes ainsi en mesure de contrôler l'ampleur de la diversité introduite dans la population. Nous avons également choisi de circonscrire la mutation aux individus obtenus par croisement. Ainsi, croisement et mutation sont couplés au sein d'une étape de reproduction assurant l'exploration de l'espace de recherche. Les individus parents sont conservés en l'état de telle sorte que l'on s'assure qu'aucune solution préalablement sélectionnée ne soit perdue du fait d'une modification malencontreuse.

5.3.2 Maintenir la diversité par spéciation

Nous étudions à présent les méthodes de *spéciation* dites de *niching*. Ces dernières tiennent leur nom du processus par lequel une seule espèce peut se différencier et donner naissance à plusieurs espèces différentes occupant des niches écologiques distinctes. Dans le cadre des algorithmes évolutionnaires, les niches sont analogues aux maxima de la fonction de performance. Le but des méthodes de *niching* est de maintenir des individus au sein de différentes niches afin d'empêcher que la population entière ne se concentre trop rapidement dans une niche correspondant à un maximum local. Trois stratégies globales de *niching* peuvent être considérées : la reproduction restreinte, le partage de ressources et le remplacement restreint.

La reproduction restreinte Lorsque les couples de parents sont formés sans tenir compte des caractéristiques des individus, il arrive fréquemment que les croisement soient *létaux*. Ce terme signifie que les descendants ne sont pas assez performants pour se maintenir à la génération suivante. Cela est par exemple le cas lorsque deux individus appartenant à des niches différentes sont recombinés. Ils sont susceptibles d'avoir une descendance peu performante, située en dehors de tout optimum. *A contrario*, deux individus appartenant à la même niche ont de fortes chances d'avoir une descendance qui leur ressemble, appartenant à la même niche. Bien sûr il s'agit là d'une généralité, la ressemblance entre parents et enfants dépendant de la méthode de recombinaison utilisée. De même, l'évolution des écarts de performance entre des individus selon

qu'ils sont similaires ou différents dépend de la fonction de performance et des briques de base utilisées pour le codage.

La *reproduction restreinte* proposée par Booker [Boo85] est supposée encourager la spéciation en limitant le nombre de croisements létaux. Elle implique que seuls les individus *similaires* puissent se reproduire. Cette ressemblance entre individus peut se fonder sur des distances génotypiques ou phénotypiques. Dans le premier cas, c'est la ressemblance entre les chromosomes codant les deux solutions parents qui est prise en compte. Dans le second, on s'intéresse à la ressemblance des solutions parents elles-mêmes. Le principal inconvénient de cette approche vient du fait que pour limiter le nombre de croisements létaux, elle limite également la production de nouvelles solutions candidates originales. L'exploration de l'espace de recherche est donc moins efficace.

Le partage de ressources Les méthodes de *fitness sharing* (ou *partage de ressources*) [GR87, HSL+02] sont fondées sur l'idée que des individus appartenant à la même niche écologique doivent partager leurs ressources. Dans le cadre des algorithmes génétiques, la ressource d'un individu correspond à sa performance. Cette méthode, décrite par Goldberg et Richardson [GR87], consiste à diminuer la performance d'un individu en fonction de sa distance avec les individus appartenant à la même niche. Par conséquent, lorsqu'une région de l'espace de recherche est densément peuplée, il est plus difficile d'y ajouter de nouveaux individus. Ces derniers auront une performance trop faible pour survivre aux générations suivantes. Les régions peu peuplées apportent au contraire un avantage sélectif aux individus qui s'y trouvent, ces derniers n'ayant pas à partager leur performance. Cette approche favorise donc la répartition de la population dans différentes régions de l'espace de recherche. Son utilisation présente cependant plusieurs difficultés. Il faut tout d'abord disposer d'une mesure de similarité entre les individus. Typiquement, on utilise la distance de Hamming, qui mesure le nombre de bits distincts entre deux chaînes binaires. Ensuite, il faut définir une fonction de partage afin de réduire la performance d'un individu en fonction de son voisinage. Enfin, il faut déterminer la taille des niches. Pour cela, on définit généralement un *rayon d'exclusion*, qui est la distance au-delà de laquelle on estime que deux individus ne peuvent appartenir à la même niche. Le choix de ce paramètre est fondamental. S'il est trop petit, on réduit notablement l'exploration de l'espace de recherche en accentuant la recherche locale. Inversement, s'il est trop grand, on ne parvient pas à discriminer les différentes niches au sein de l'espace de recherche et la finalité même de la méthode est remise en cause. Deb et Goldberg [DG89] ont proposé une méthode permettant d'estimer le rayon d'exclusion. Elle se fonde sur l'hypothèse que le nombre de niches est connu et qu'elles sont dispersées au sein de l'espace de recherche (et non concentrées dans une même région). Lorsqu'il existe de nombreux maxima au voisinage du maximum global, les méthodes de *fitness sharing* perdent leur efficacité [GDH92, SK98].

Le remplacement restreint De Jong [DJ75] a proposé une méthode de remplacement restreint appelé *crowding*. L'idée consiste à tirer au hasard un petit nombre d'individus (de l'ordre de deux ou trois) dans la population afin de les comparer à un descendant à l'issue de la reproduction. Le descendant remplace l'individu qui lui ressemble le plus d'après la distance de Hamming. Cette méthode est assez inefficace car elle réalise un grand nombre d'erreurs de remplacement [3] et on lui préfère généralement le *fitness sharing* [DG89]. En effet, le *crowding* ne tenant pas compte de la performance des individus, un enfant peut remplacer un individu nettement plus performant.

[3] Une erreur de remplacement survient lorsqu'un individu performant est éliminé au profit d'un individu de moins bonne qualité

Harik [Har95] a proposé une méthode de *sélection par tournoi restreint* qui corrige cette erreur. De nouveau, un petit nombre d'individus sont tirés au hasard pour être comparés à un descendant. Cependant, ce dernier rentre en compétition avec l'individu qui lui ressemble le plus au lieu de le remplacer automatiquement. Le remplacement n'a lieu que si l'enfant est plus performant que son compétiteur. Bien qu'elle soit plus performante que le *crowding*, cette méthode souffre du même inconvénient que le *fitness sharing* : il faut déterminer le nombre d'individus auquel un enfant doit être comparé. Ce paramètre — généralement appelé *facteur de peuplement* — est analogue au rayon d'exclusion, et détermine dans une large mesure la taille des niches que l'on prend en compte.

Une approche alternative et beaucoup plus simple est le *deterministic crowding* (DC) introduit par Mahfoud [Mah95]. Cette technique consiste à comparer les deux enfants produits par recombinason avec les deux parents recombinés. Chaque enfant rentre en compétition avec l'un des deux parents et ne le remplace que s'il est plus performant. L'appariement entre enfants et parents est réalisé de la manière suivante :
Soit $P1$, $P2$, $E1$ et $E2$ le premier et le deuxième parent, le premier et le deuxième enfant respectivement. Soit $d(X, Y)$ une distance entre deux individus X et Y. Si $d(P1, E1) + d(P2, E2) < d(P1, E2) + d(P2, E1)$ alors $P1$ rentre en compétition avec $E1$ et $P2$ rentre en compétition avec $E2$. Dans le cas contraire, $P1$ rentre en compétition avec $E2$ et $P2$ rentre en compétition avec $E1$.
Comme on peut le constater, cette technique ne requiert aucun paramètre supplémentaire. Elle offre de bonnes performances, comparables à celles de la *sélection par tournoi restreint* [SK98] tout en étant plus simple à mettre en œuvre.

Choix d'un méthode de spéciation Dans ces travaux, nous avons choisi d'utiliser la méthode du *deterministic crowding* (DC). Celle-ci ne restreint pas le champ de la reproduction. Elle permet de recombiner des solutions dissemblables et de générer des solutions originales. Il s'agit d'une méthode générale dont l'élaboration et le fonctionnement ne sont pas dépendantes du codage ou du paysage de la fonction de performance. Notamment, contrairement aux méthodes de *fitness sharing* présentées plus haut, elle ne fait pas d'hypothèses concernant le nombre ou la position relative des maxima de la fonction de performance dans l'espace de recherche. Enfin, notons que le DC s'accorde aisément avec les choix que nous avons faits jusqu'à présent. Dans la sous-section 5.2.1, nous avons expliqué vouloir appliquer une stratégie de remplacement élitiste dans le cadre d'un algorithme à états stationnaires. Il se trouve que le DC est une stratégie de remplacement élitiste : parmi les $M + 2$ individus dont on dispose à l'issue de la reproduction, le meilleur est systématiquement conservé jusqu'à la génération suivante. Par ailleurs, selon Watson et Pollack [WP00], pour que le DC fonctionne bien, les individus recombinés doivent être choisis de manière aléatoire comme nous l'avons décidé dans la sous section 5.2.1. Pour finir, dans notre algorithme, la distance utilisée pour calculer les distances relatives entre enfants et parents est la distance de Hamming.

5.3.3 Synopsis des algorithmes génétiques à états stationnaires

À ce stade, les grandes lignes de l'algorithme évolutionnaire que nous allons utiliser peuvent être décrites de la façon suivante :
L'algorithme est initialisé aléatoirement. Au début de chaque génération, deux parents sont choisis au hasard puis sont croisés avec l'une des trois méthodes de recombinaison suivantes : recombinaison classique, recombinaison parentale ou recombinaison relationnelle. À l'issue d'un croisement, les deux enfants produits sont réparés puis éventuellement soumis à un opérateur de mutation. Leur performance est ensuite évaluée au moyen du score BIC et une méthode

de remplacement est utilisée : soit on recourt à un remplacement élitiste, soit on utilise le *deterministic crowding*. Le processus est répété d'une génération sur l'autre jusqu'à convergence. Nous avons choisi de considérer que la convergence est atteinte lorsqu'aucune amélioration n'est observée durant un millier d'itérations.

Initialisation de la population : un ensemble de DAG est généré aléatoirement. Tous sont évalués grâce au score BIC.

Répéter jusqu'à ce qu'un critère d'arrêt soit satisfait :

1. Tirer aléatoirement deux DAG parents au sein de la population : $P1$ et $P2$.
2. $P1$ et $P2$ sont recombinés pour produire deux enfants : $C1$ et $C2$ grâce à l'une des trois méthodes suivantes :
   – recombinaison classique ;
   – recombinaison parentale ;
   – recombinaison relationnelle.
3. $C1$ et $C2$ sont ensuite réparés pour respecter la contrainte d'acyclicité ainsi que la contrainte sur le degré entrant maximum des sommets.
4. $C1$ et $C2$ subissent des additions et des suppressions aléatoires d'arcs sous l'effet d'un opérateur de mutation.
5. Le score BIC des DAG $C1$ et $C2$ est calculé.
6. On élimine deux individus de la population courante pour restaurer la taille initiale de la population :
   – remplacement élitiste : $C1$ et $C2$ remplacent les deux individus les moins performants de la population si leurs scores sont plus élevés ;
   – *deterministic crowding* : $C1$ et $C2$ sont appariés avec $P1$ et $P2$ suivant la distance de Hamming. Au sein de chaque couple parent-enfant, celui dont le score est le plus élevé remplace l'autre.

Dans le chapitre suivant, nous allons tester cet algorithme pour les différents méthodes de recombinaison et de maintien de la diversité que nous avons sélectionnées. Plus précisément, nous allons comparer les recombinaisons classique, parentale et relationnelle en fonction de l'utilisation de la mutation et du DC. La mutation et le *deterministic crowding* prenant place dans deux phases distinctes de l'algorithme génétique, nous allons comparer les résultats obtenus avec et sans mutation d'une part, et avec et sans DC d'autre part. Lorsque le DC n'est pas utilisé, les descendants remplacent systématiquement les deux individus les moins bons de la population s'ils s'avèrent meilleurs (stratégie de remplacement élitiste).

Chapitre 6

Résultats numériques

Nous présentons à présent les différents tests que nous avons effectués afin d'analyser les mérites des méthodes retenues dans le chapitre précédent. Les plus performantes seront validées par comparaison avec des algorithmes d'apprentissage classiquement utilisés dans la littérature.

6.1 Méthodes de validation et d'évaluation

Afin d'évaluer les performances et les propriétés d'un algorithme d'apprentissage de structure, nous devons mesurer sa capacité à retrouver la véritable structure du réseau de régulation étudié. Actuellement, il n'existe aucun standard incluant des données statiques expérimentales et une connaissance complète des régulations génétiques sous-jacentes susceptibles de servir de référence commune pour l'évaluation d'algorithmes d'apprentissage. Lorsqu'une nouvelle approche est introduite, il est intéressant d'utiliser des données artificielles pour tester l'algorithme dans diverses conditions. C'est ce qui a été réalisé par Smith et col. [SJH02], qui ont conçu un simulateur permettant de générer des données dynamiques représentant un système biologique complexe intégrant entre autres, l'expression de gènes chez l'oscine (oiseau chanteur). Mendes et collègues [MSY03] ont également proposé un système permettant de générer aléatoirement des réseaux de gènes artificiels. Ces derniers reposent sur des modèles dynamiques non linéaires et sont utilisés pour générer des données simulant des données de puces à ADN. Werhli et collègues [WGH06] se sont appuyés sur un réseau de signalisation cellulaire décrivant les interactions entre 11 phospholipides et protéines phosphorylées du système immunitaire humain pour mener une étude comparative entre différentes méthodes d'apprentissage. Ils ont utilisé des données expérimentales de cytométrie mais aussi des données simulées obtenues à partir du réseau reconstruit par les experts.

Pour notre part, nous avons choisi de considérer le modèle synthétique proposé dans [LBU04], qui est un réseau Bayésien bio-réaliste fondé sur des connaissances établies concernant le réseau de régulation de l'insuline comprenant 35 variables (voir la figure 6.1). Nous avons généré des échantillons de taille croissante pour les 35 variables du modèle. Ce modèle présente l'avantage d'être discret et statique, il correspond donc au type de données que nous avons proposé d'utiliser pour l'apprentissage de réseaux de régulation au chapitre 3. En outre, son utilisation permet de respecter le postulat causal de Markov et l'hypothèse de fidélité causale puisque le modèle objectif est bien un réseau Bayésien et que toutes les variables du problème sont prises en compte. Bien sûr, ce modèle est moins réaliste que les modèles décrits précédemment. Nous pensons cependant que dans le cadre que nous nous sommes fixé, être en mesure de maîtriser ce problème d'apprentissage est une étape fondamentale avant d'envisager d'apprendre des modèles

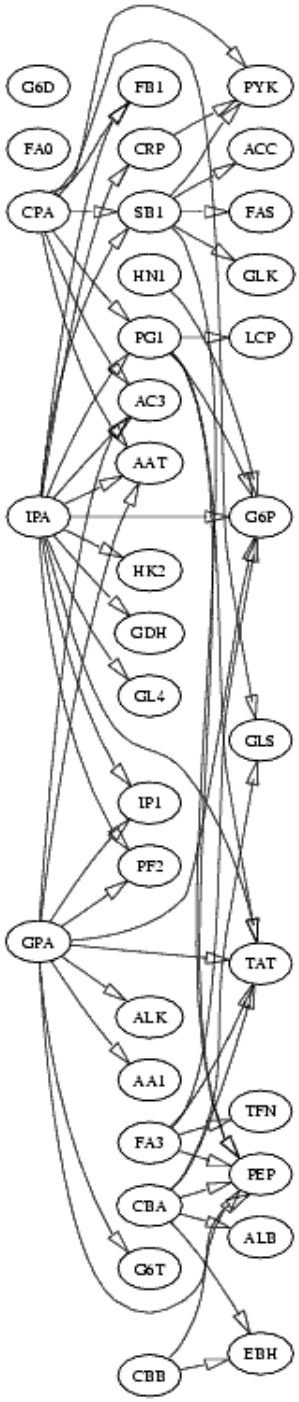


Fig. 6.1 – Réseau Insuline (Le, Bahl et Ungar ; In Silico Biology, 2004).

plus complexe.

Un autre point fondamental dans la problématique de reconstruction des réseaux de régulation génétique tient à la quantité des données disponibles. Comme nous l'avons démontré dans le second chapitre, il existe une grande variété de puces à ADN ainsi que de nombreuses façons de concevoir une étude de profil d'expression pour un type de puces donné. Cela explique que malgré le nombre sans cesse croissant de données de profils d'expression publié dans la littérature, il soit si difficile de construire de grandes bases d'apprentissage. En effet, les jeux de données produits en laboratoire sont généralement de taille limitée. Du fait du caractère hétérogène de ces jeux de données il n'est généralement pas possible de les utiliser conjointement. Il faut donc se résoudre à n'utiliser qu'un nombre restreint d'observations. En consultant les dépôts de données tel que *Gene Expression Omnibus* ou *Array Express* on peut trouver des jeux de données comptant quelques centaines d'observations. Il nous semble donc raisonnable d'envisager l'apprentissage de structure de réseaux Bayésiens à partir de 300 ou 400 observations, bien que cela reste un nombre de données très généreux. Enfin, le dernier point concerne la reproductibilité des résultats. Afin de nous assurer que les performances d'un algorithme d'apprentissage sont indépendantes de la base d'apprentissage choisie, chaque test sera répété sur des jeux de données synthétiques distincts et indépendants.

Dans ce qui suit, nous allons donc évaluer les méthodes d'apprentissage étudiées en nous appuyant sur des jeux de données artificiels de taille réduite, indépendants, générés à partir d'un modèle biologiquement pertinent issu de la littérature.

6.2 Mesures utilisées pour l'analyse des résultats

Avant de définir les indices d'évaluation permettant de mesurer les performances de l'apprentissage, il nous faut noter que le BIC est un critère Markov équivalent : il attribue le même score à tous les DAG appartenant à la même classe d'équivalence Markovienne [VP91]. Le processus d'apprentissage ne peut donc pas discriminer deux DAG appartenant à la même classe d'équivalence Markovienne sur la seule base des données observées. C'est pourquoi les analyses ont été réalisées sur le CPDAG correspondant au DAG obtenu à l'issue de chaque apprentissage. L'évaluation du processus d'apprentissage a donc consisté en la comparaison du CPDAG appris avec le CPDAG correspondant au modèle de référence.

Afin d'évaluer la qualité d'un algorithme d'apprentissage nous souhaitons étudier sa capacité à identifier le modèle dont les données sont issues. Nous proposons donc d'étudier la ressemblance entre le CPDAG appris et le CPDAG objectif au moyen des indices suivants.

Le nombre de vrais positifs Un vrai positif (tp) est un arc qui apparaît à la fois dans le graphe appris et dans le graphe de référence. S'il est orienté dans le graphe appris, il doit l'être également, et dans la même direction, dans le graphe de référence. Un arc non orienté dans le graphe appris sera considéré comme un vrai positif même s'il est orienté dans le graphe de référence, dans la mesure où l'apprentissage ne fait aucune spéculation concernant son orientation.

Le nombre de vrais négatifs Un vrai négatif (tn) se caractérise par l'absence d'arcs entre deux nœuds donnés, à la fois dans le graphe appris et le graphe de référence.

Le nombre de faux positifs Il est égal au nombre total de positifs (le nombre d'arcs dans le graphe appris) auquel on retranche le nombre de vrais positifs.

Le nombre de faux négatifs Il est égal au nombre total de négatifs (le nombre de paires de sommets non connectés) auquel on retranche le nombre de vrais négatifs.

Au final, les CPDAG appris peuvent être évalués à l'aide des mesures suivantes.

Sensibilité Également appelée *précision*, la *sensibilité* est égale à $\frac{tp}{tp+fn}$.

Valeur de prédiction positive Généralement notée *ppv*, *valeur de prédiction positive* est égale à $\frac{tp}{tp+fp}$.

Spécificité Également appelée *rappel*, la *spécificité* est égale à $\frac{tn}{tn+fp}$.

Toutefois dans cette étude, la *spécificité* apparaît comme une métrique peu pertinente. L'introduction d'une borne supérieure sur le degré entrant des nœuds du réseau, au même titre que la contrainte sur la complexité présente au sein du BIC, garantissent la génération de solutions parcimonieuses. Dans la mesure où le graphe de référence comme les graphes appris présentent un faible nombre d'arcs, le nombre de vrais négatifs est toujours élevé en comparaison du nombre total de négatifs ($tn + fp$), donc la *spécificité* n'est pas discriminante.

Le temps de calcul ne nous a pas semblé être un critère d'évaluation fondamental pour cette étude. Lorsque nous réalisons une comparaison entre différentes méthodes d'apprentissage, il est certes naturel de mettre en relief leurs temps de calcul respectifs. Dans le cas d'heuristiques stochastiques ces derniers sont généralement élevés. Sous Matlab, plusieurs heures sont nécessaires pour exécuter un algorithme génétique ou un algorithme MCMC sur des systèmes comportant quelques dizaines de variables. Cependant, comparé au temps nécessaire aux biologistes pour générer des données, cela semble négligeable. En effet, après plusieurs mois d'expérimentation, le fait qu'un algorithme mette quelques minutes ou quelques heures pour proposer une hypothèse de réseau de régulation à partir des données obtenues importe peu. D'autant plus que le temps nécessaire à la validation des résultats et à leur interprétation peut également nécessiter plusieurs semaines. Typiquement, les influences régulatrices représentées au sein de la structure du modèle appris doivent être confrontées à la littérature et les plus intéressantes doivent être testées au moyen d'expériences supplémentaires. Il est donc possible de se montrer plus tolérant face au temps de calcul conséquent des algorithmes évolutionnaires que dans d'autres domaines d'applications.

Nous avons donc choisi d'évaluer la qualité des différentes approches d'apprentissage en nous référant à la *sensibilité* et la *ppv* des structures de modèles appris. Bien sûr, l'inconvénient majeur d'un algorithme stochastique réside dans la variabilité des solutions qu'il propose à l'issue de différentes exécutions. Tous les tests ont donc été répétés afin de rendre compte du comportement moyen des algorithmes étudiés. Au final, nous souhaitons apprendre des réseaux fidèles à l'original ayant une bonne *sensibilité* (capacité à découvrir des interactions) mais aussi et surtout une bonne *ppv* afin de limiter les faux positifs. En effet, une sensibilité élevée, si elle s'accompagne de nombreux faux positifs, est de peu d'intérêt car les interactions proposées par le modèle étant fausses pour la plupart, il est difficile de les tester et de les confirmer au moyen d'expériences complémentaires. S'il est indéniable que les biologistes sont prêts à accepter une faible proportion d'erreurs afin de faire des découvertes, il est fondamentale que ces dernières ne soient pas noyées parmi les faux positifs. Comme nous le verrons par la suite les méthodes d'apprentissage tendent à founir un taux élevé de faux positifs.

6.3 Comparaison de différentes approches évolutionnaires

Dans un premier temps, nous étudions les performances de l'algorithme évolutionnaire décrit précédemment selon la stratégie de recombinaison et la méthode de préservation de la diversité

utilisées. Tout d'abord, nous avons comparé l'effet de la mutation et celui du *deterministic crowding* pour différentes stratégies de reproduction. Plus précisément, nous avons étudié le comportement des trois stratégies de reproduction présentées au chapitre précédent, en faisant varier le taux d'échange pour les recombinaisons relationnelles et parentales. Ce taux d'échange permet de paramétrer le nombre de gènes virtuels échangés entre deux chromosomes par la recombinaison uniforme. Puisque nous n'étions pas intéressé par un réglage précis de ce paramètre, nous avons simplement considéré un taux d'échange bas $(0,1)$ et élevé $(0,4)$. Nous avons également considéré une faible probabilité de mutation de $0,002$ impliquant de l'ordre de 2 arcs modifiés par DAG. Nous avons travaillé sur des populations de taille relativement limitée (en comparaison de la taille de l'espace des solutions) de 200 DAG. Enfin, l'algorithme s'arrête lorsque le meilleur score de la population ne montre aucune amélioration durant au moins 1 000 itérations. Nous avons en plus imposé un nombre maximum de 50 000 itérations. Nous rappelons que cet algorithme ne produit que deux solutions candidates par itération, c'est pourquoi ces dernières sont si nombreuses.
Chaque test a été effectué 10 fois, en s'appuyant sur des jeux de données distincts et indépendants pour juger de la robustesse des différentes approches évolutionnaires. Pour tenir compte de la disponibilité des données biologiques (qui sont des instantanés de l'activité transcriptionnelle des cellules) nous avons considéré des échantillons de faible taille (300 mesures). Nous considérerons cependant des tailles d'échantillon variables dans un second temps.

La *sensibilité* et la *ppv* que nous avons obtenues pour chaque test sont représentées dans les tables 6.1 et 6.2, respectivement. Ces résultats correspondent à la moyenne et à l'écart-type de chacun de ces indices de qualité sur les 10 exécutions réalisées pour chaque test. Pour favoriser la lisibilité, ces résultats ont été exprimés en termes de pourcentage et arrondis à l'entier le plus proche.
Les lignes correspondent aux stratégies de recombinaison : recombinaison relationnelle (lignes 1 et 2), recombinaison parentale (lignes 3 et 4), recombinaison classique (lignes 5). Pour les recombinaisons relationnelles et parentales, deux lignes sont disponibles puisque le croisement uniforme sur lequel elles reposent est testé pour un taux d'échange élevé (lignes 1 et 3) et faible (lignes 2 et 4). Les colonnes correspondent aux diverses techniques de préservation de la diversité utilisées dans chaque test. Nous comparons les cas où aucune de ces techniques n'est utilisée (colonne 1), où seulement l'une des deux est utilisée (colonne 2 pour la mutation et colonne 3 pour DC) et où les deux sont utilisées simultanément (colonne 4).

Lorsque nous considérons la première colonne des tables 6.1 et 6.2, nous constatons qu'en l'absence de méthode de préservation de la diversité (colonne 1), deux tendances émergent. En premier lieu, pour les recombinaisons relationnelles et parentales, les résultats s'avèrent meilleurs pour un taux d'échange élevé. Cela était prévisible dans la mesure où un taux d'échange plus élevé favorise un mélange plus important des gènes virtuels entre modèles candidats. L'exploration de l'espace de recherche s'en trouve accélérée du fait de la génération d'une plus grande variété de modèles candidats. Dans un second temps, si l'on considère le taux d'échange élevé, la recombinaison relationnelle surpasse la recombinaison parentale qui, à son tour, fonctionne nettement mieux que la recombinaison classique. En effet, la recombinaison relationnelle pratique la recombinaison à un niveau plus fin (interactions élémentaires) que les deux autres qui manipulent d'un bloc de larges sous-ensembles d'(in)dépendances conditionnelles. Cela permet à l'algorithme génétique de s'échapper plus facilement d'un optimum local pour atteindre de meilleures régions de l'espace de recherche avant d'être piégé par l'homogénéisation prématurée de l'algorithme. Cela laisse également supposer que notre hypothèse concernant les briques de

Tab. 6.1 – Moyenne ± écart-type de la **sensibilité** (×100) : comparaison de différentes stratégies d'évolution - 10 exécutions. DC = Deterministic Crowding ; Mut = Mutation ; NoDC = pas de Deterministic Crowding ; NoMut = pas de Mutation.

Recombinaison	NoDC/NoMut	NoDC/Mut	DC/NoMut	DC/Mut
Recomb. Relationnelle - Élevée	43 ± 4	61 ± 6	63 ± 3	68 ± 4
Recomb. Relationnelle - Faible	18 ± 7	42 ± 8	68 ± 4	68 ± 4
Recomb. Parentale - Élevée	23 ± 7	56 ± 7	48 ± 3	66 ± 4
Recomb. Parentale - Faible	12 ± 5	33 ± 6	61 ± 4	60 ± 2
Recomb. Classique	12 ± 4	43 ± 7	43 ± 5	59 ± 7

Tab. 6.2 – Moyenne ± écart-type de la **PPV** (×100) : comparaison de différentes stratégies d'évolution - 10 exécutions. DC = Deterministic Crowding ; Mut = Mutation ; NoDC = pas de Deterministic Crowding ; NoMut = pas de Mutation.

Recombinaison	NoDC/NoMut	NoDC/Mut	DC/NoMut	DC/Mut
Recomb. Relationnelle - Élevée	61 ± 12	58 ± 8	84 ± 5	74 ± 8
Recomb. Relationnelle - Faible	18 ± 8	22 ± 8	82 ± 9	80 ± 6
Recomb. Parentale - Élevée	26 ± 8	38 ± 4	68 ± 10	69 ± 6
Recomb. Parentale - Faible	12 ± 5	14 ± 4	79 ± 6	63 ± 7
Recomb. Classique	12 ± 5	21 ± 4	62 ± 7	52 ± 11

base des solutions recherchées est soit fausse soit mal exploitée par les méthodes de recombinaison proposées. Dans le cas de la recombinaison parental, cela peut s'expliquer par le fait que les parentés des sommets du DAG qui constituent nos briques de base n'évoluent pas ou très peu sous l'effet de la recombinaison (la réparation apportant quand même quelques modifications).

En comparant les colonnes 1 et 2 des tables 6.1 et 6.2, il apparaît que les précédentes observations concernant la comparaison des stratégies de recombinaison demeurent valides en présence de l'opérateur de mutation. Cependant, alors que la *ppv* (table 6.2) reste stable, la *sensibilité* (table 6.1) augmente significativement pour toutes les stratégies de recombinaison. L'amélioration apportée par la mutation est particulièrement importante pour les méthodes de recombinaison les moins efficaces. En effet, la mutation modifie les larges sous-chaînes échangées entre chromosomes classiques et les listes parentales constituant les gènes virtuels des chromosomes parentaux. Elle permet donc à l'algorithme de s'échapper des minima locaux où ces méthodes de recombinaison ont tendance à converger du fait de leur nature conservatrice. De la même manière, on remarquera que pour les recombinaisons fondées sur le croisement uniforme, cette amélioration est plus importante pour un faible taux d'échange. La mutation permet d'obtenir de nouvelles topologies de graphes qui n'auraient pu être prises en considération du fait de la lenteur du mélange des gènes virtuels résultant de ce paramétrage. Il est par contre plus difficile d'expliquer pourquoi la mutation améliore plus particulièrement la *ppv*. L'hypothèse la plus probable est que du fait de dépendances fortes entre les gènes virtuels, il est difficile d'identifier de nouveaux arcs du DAG de référence au moyen de modifications élémentaires des DAG candidats. Il peut être nécessaire de considérer l'introduction simultanée de plusieurs arcs ayant un rôle commun dans le modèle (représentant des co-régulateurs d'un gène par exemple) pour améliorer le score. Inversement, il est plus facile d'éliminer un arc surnuméraire, sa suppression devant se traduire par une amélioration de la performance du DAG muté.

Une simple comparaison entre les colonnes 2 et 3 nous montre que remplacer la mutation par le DC améliore fortement les résultats. On constate une augmentation de la *sensibilité* pour les recombinaisons relationnelles et parentales ayant un faible taux d'échange alors que les autres stratégies de recombinaison conservent des résultats similaires. Cependant, concernant la *ppv*, toutes les stratégies de recombinaison présentent une importante amélioration. De nouveau, ce sont les méthodes de croisement les moins performantes qui bénéficient le plus de l'apport du DC. Cette amélioration tend à effacer le différentiel de performances précédemment constaté entre ces différentes stratégies de recombinaison : lorsque le DC est appliqué, elles donnent toutes des résultats satisfaisants. En l'occurrence, cette amélioration s'explique par la capacité du DC à retarder l'homogénéisation de la population afin de prévenir la convergence prématurée de l'algorithme. Cela permet aux méthodes les moins performantes en termes de découverte de nouvelles solutions candidates de poursuivre leur recherche vers de nouveaux minima.

Finalement nous avons étudié l'effet conjoint des deux techniques de préservation de la diversité en comparant les colonnes 3 et 4. Étonnamment, alors que la mutation n'apporte qu'une augmentation modérée de la *sensibilité* lorsqu'elle est ajoutée au DC, nous observons également une diminution de la *ppv* pour certaines méthodes de recombinaison. Cependant, ces remarques reposent surtout sur des tendances puisque, considérant la forte variabilité des résultats, la plupart de ces variations ne sont pas significatives.

Il doit être noté que les différentes approches évolutionnaires que nous venons de comparer ne réalisent pas le même nombre d'évaluations de la fonction objectif avant de converger. La princi-

pale différence est imputable au *niching*. Dans nos expériences, un AE utilisant la recombinaison relationnelle, un taux d'échange élevé, une faible probabilité de mutation et le *deterministic crowding* réalise entre 40 000 et 50 000 évaluations du score BIC. Le même algorithme, privé de *deterministic crowding* réalise entre 20 000 et 30 000 évaluations du score BIC. En somme, la spéciation multiplie par deux le coût en termes de calcul. Cela n'est pas négligeable mais dans la mesure où les valeurs présentées dans les deux cas sont du même ordre, nous estimons que les comparaisons réalisées sont justes. En effet, en l'absence de mécanisme de spéciation, recourir à une initialisation multiple n'a que peu d'intérêt : il suffit de répéter l'exécution de l'algorithme privé du *niching* une seule fois pour atteindre le même coût de calcul que lorsque le *niching* est utilisé.

Nous montrons qu'un taux de mutation modéré est nécessaire au bon fonctionnement d'un algorithme génétique reposant sur un schéma de sélection élitiste. Cependant, le recours au *deterministic crowding* — qui améliore sensiblement le processus d'apprentissage — contrebalance nettement l'absence totale de mutation. Par la suite, nous utiliserons conjointement ces deux techniques. La recombinaison relationnelle avec un taux d'échange élevé est l'approche qui a donné les meilleurs résultats à travers les différents tests. Nous avons donc choisi de nous appuyer sur une recombinaison relationnelle ayant un taux d'échange élevé, ainsi que sur l'utilisation conjointe de la mutation et du *deterministic crowding* pour la suite de ce chapitre.

6.4 Visualiser l'effet du Deterministic Crowding sur la répartition des DAG

Le comportement des algorithmes évolutionnaires est généralement analysé à travers le suivi au cours du temps d'une métrique caractérisant les individus de la population : le plus souvent, il s'agit du meilleur score obtenu parmi les individus de la population rendant compte des propriétés de convergence de l'algorithme. Pour étudier plus précisément l'effet du *niching* sur le fonctionnement d'un AE, nous avons besoin de suivre les modifications survenant au sein de la population au cours du processus d'évolution. Il nous faut donc être capables de comparer les structures candidates elles-mêmes et pas seulement leurs indices de qualité tels que la *ppv* ou la *sensibilité*. Nous proposons de visualiser des graphes comme des points dans un espace à deux dimensions afin de mettre en lumière l'évolution de leur distribution au sein de la population.

6.4.1 Visualisation par Sammon-mapping

L'impact du *niching* sur l'exploration de l'espace des DAG peut être étudié en visualisant la façon dont les structures des solutions candidates se distribuent au sein des populations successives. Pour cela, nous utilisons le Sammon-mapping [Sam69], un algorithme permettant de projeter des données issues d'un espace de dimension e dans un espace de dimension f tel que $f < e$. Durant la projection, le Sammon-mapping tend à préserver les distances entre tous les points du jeu de données pris deux à deux afin de préserver la structure des données.

Considérons un ensemble de N objets. Chaque objet est représenté par un point dans un espace à e dimensions. Le but du Sammon-mapping est de trouver N points dans un espace de dimension f inférieur à e, de telle sorte que les distances correspondantes soient proches des distances au sein de l'espace d'origine :

- $d_{ij}, \quad \forall i, j = 1, \cdots, N$ est la distance entre deux points dans l'espace à e dimensions appelé espace d'origine.

– $\delta_{ij}, \quad \forall i, j = 1, \cdots, N$ est la distance entre deux points dans l'espace à f dimensions appelé espace d'arrivé.

Typiquement, afin de faciliter la visualisation, nous avons $f = 2$, auquel cas δ_{ij} est la distance euclidienne entre les points i et j dans un espace 2D. Afin d'identifier les N points au sein de l'espace d'arrivé dont les distances deux à deux approchent au mieux celles dans l'espace d'origine, on utilise un critère de coût. Ce dernier est généralement nommé « fonction de stress » et est noté S. Il mesure l'écart entre les configurations dans les deux espaces :

$$S = \frac{1}{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\delta_{ij}} \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}$$

On cherche donc à identifier les N points au sein de l'espace d'arrivé qui minimisent cette fonction de stress.

Dans notre cas, les objets traités sont des graphes que nous souhaitons représenter par des points dans un espace 2D. Il nous faut donc définir une distance entre ces graphes. Nous utilisons une distance de Hamming afin de comparer les matrices d'adjacence codant les différents DAG de la population. Cette distance repose sur le comptage des différences survenant au sein de ces matrices qui pour l'occasion sont reprises sous la forme de vecteurs binaires.
Soit A la matrice d'adjacence codant la structure de réseau Bayésien $G_k \in \mathcal{G}$:

$$A = \{A_{ij}\} \quad \text{avec} \quad i, j \in \{1, 2, \ldots, n\} \quad \text{et} \quad A \in \{0, 1\}^{n^2}$$

On considère V_k un vecteur correspondant à la matrice d'adjacence A :

$$V = \{V_k\} \, \forall k \in \{1, \ldots, n^2\} \, | \, V_k = A_{ij}$$

La distance de Hamming entre deux vecteurs V_1 et V_2 codant deux DAG G_1 et G_2 est donnée par :

$$H(V_1, V_2) = \sum_{i=1}^{n^2} \otimes(V_1^k, V_2^k), \forall V_1, V_2 \in \{0, 1\}^{n^2}$$

où $\otimes(a, b)$ est la fonction XOR. Dit plus simplement, la distance de Hamming compte le nombre de bits distincts entre deux vecteurs binaires.

Nous avons utilisé le *Sammon-mapping* pour représenter dans un espace 2D les populations d'un algorithme évolutionnaire enregistrées toutes les 10 000 générations. À chaque fois, nous comparons la population à la génération g avec les populations enregistrées aux générations $g + 10\ 000$ à 50 000. Rappelons qu'au-delà de 50 000 générations, l'algorithme génétique est stoppé. Nous comparons l'effet du *niching* sur la répartition des individus de la population au fil des générations en fonction de deux méthodes de recombinaison : la recombinaison parentale et la recombinaison relationnelle. Pour éviter que la mutation n'interfère avec le *niching* quant au contrôle de l'homogénéisation, celle-ci a été désactivé pour ces tests.

Nous commençons par étudier l'évolution de la population d'un AE utilisant la recombinaison relationnelle avec un fort taux d'échange. Les résultats sont présentés à la figure 6.2. Le *niching* est utilisé dans la colonne de gauche et non dans la colonne de droite. Considérons tout d'abord la sous-figure A1. Dans la mesure où ce sont les distances relatives entre matrices d'adjacence qui sont représentées, un effet artefactuel, que nous appelons l'effet « donut », apparaît. Si nous

avions représenté uniquement la population initiale (Gen. 0), nous aurions obtenu un disque représentant des individus distribués uniformément, ces derniers étant générés aléatoirement. La comparaison de deux populations appartenant aux générations 0 et 10 000-50 000 montre une population répartie sur un anneau (population à la génération 0) entourant de manière plus ou moins régulière une population répartie sur un disque de rayon inférieur (population aux générations 10 000-50 000). Cela illustre le phénomène d'homogénéisation : les individus se ressemblent de plus en plus au cours du temps et leurs distances relatives tendent donc à se resserrer. La densité des individus (i.e. leur répartition au sein de l'espace de représentation 2D) est donc de plus en plus marquée. Les individus des générations 10 000-50 000 ayant eu le temps de converger vers une région plus restreinte de l'espace de recherche, les individus de la génération 0 sont nettement plus différents de ces derniers qu'ils ne le sont entre eux. On remarque que 10 000 générations plus tard (sous-figure A2), l'homogénéisation est complète. Les individus de la génération 10 000 sont répartis uniformément autour des populations des générations 20 000 à 50 000 qui sont déjà concentrées en un point. À la génération 20 000, l'algorithme a convergé, tous les individus de la population sont identiques. Si on regarde la colonne de droite, on remarque qu'en l'absence de *niching* le phénomène d'homogénéisation est accentué : la population est concentrée en un seul point avant 10 000 générations.

Nous étudions à présent l'effet du *niching* sur le comportement de la population dans un AE utilisant la recombinaison parentale (figure 6.3). Afin de nous placer dans une situation radicalement différente de la précédente, nous considérons le cas où nous avons utilisé un taux d'échange faible $(0, 1)$. Si nous regardons la colonne de droite, nous constatons que de nouveau, en l'absence de *niching*, la population a convergé en moins de 10 000 générations. Cette fois cependant, l'utilisation du *niching* (colonne de gauche) permet non seulement de ralentir considérablement l'homogénéisation de l'algorithme, mais également de structurer la population. En effet, les populations des générations 40 000 et 50 000 (sous-figure A3) ne sont plus uniformément réparties dans une région de l'espace 2D. Dans ce dernier cas, le *niching* semble être parvenu à maintenir une réelle hétérogénéité au sein de l'espace des solutions.

À l'issue de cette expérience, il apparaît clairement que la méthode de spéciation que nous avons utilisée permet de maintenir la diversité dans une population de solutions candidates. Ces résultats confirment également les performances observées des différentes méthodes de reproduction en fonction de l'utilisation du *deterministic crowding* : cette technique avantage tout particulièrement les méthodes de recombinaison qui ré-associent une faible proportion des gènes virtuels de leur parents et qui par conséquent, produisent des enfants très proches de leur parents.

6.4.2 Visualisation par analyse en composantes principales

Afin de confirmer les observations précédentes qui, nous l'avons souligné, reposent sur une représentation très contrainte de la réalité, nous proposons d'utiliser une seconde méthode de visualisation plus élaborée. Nous avons utilisé l'Analyse en Composantes Principales Kernelisée (KPCA) introduite par Schölkopf *et al.* [SSM97] pour projeter des DAG dans un espace à deux dimensions. La KPCA consiste à appliquer la PCA classique dans l'espace des caractéristiques doté d'un noyau k faisant office de produit scalaire. Un noyau k est une fonction de similarité qui est positive semi-définie. La KPCA nous fournit un moyen simple d'appliquer la PCA à des objets complexes pour lesquels une fonction noyau peut être définie.

Noyau entre graphes : le noyau du produit direct De nombreux noyaux entre graphes ont été introduits par Gartner dans [GDR03]. Plus particulièrement, il a proposé le noyau du produit direct fondé sur le produit direct de graphes. Étant donnés deux graphes orientés et

étiquetés G_1 et G_2, on définit $E_\times$ la matrice d'adjacence de leur produit direct $G_1 \times G_2$. Dans notre cas particulier, l'ensemble des sommets est toujours le même pour tous les graphes et par conséquent, la matrice $E_\times$ est le produit élément par élément des matrices d'adjacence de G_1 et G_2. Le noyau du produit direct $k_\times$ est défini comme suit :

$$k_\times(G_1, G_2) = \sum_{i,j=1}^{m} \sum_{n=0}^{\infty} \frac{\beta^n}{n!} E_\times^n = \sum_{i,j=1}^{m} \exp \beta E_\times$$

avec m le nombre de sommets des graphes, $\beta \in \mathcal{R}^+$ et exp est l'exponentiel de matrice. Nous avons choisi d'appliquer une version normalisée du précédent noyau :

$$k_\times^{norm}(G_1, G_2) = \frac{k_\times(G_1, G_2)}{\sqrt{k_\times(G_1, G_1).k_\times(G_2, G_2)}}$$

Réduire la visualisation d'un ensemble de graphes à un espace 2D implique des pertes d'informations évidentes. La forme des nuages de points notamment n'est pas interprétable. Cette visualisation fournit cependant une image intelligible de la répartition des DAG au sein d'une population ainsi que des tendances pertinentes sur le comportement de la population.

Résultats À la vue des résultats obtenus par *Sammon-mapping*, nous avons décidé de modifier un peu notre approche. Tout d'abord, nous avons choisi de considérer des intervalles de temps plus courts entre les populations comparées. C'est pourquoi durant le processus d'apprentissage, la population des structures de réseaux Bayésiens est enregistrée toutes les 2 000 générations. Par ailleurs, nous avons décidé de ne plus comparer une population donnée à l'ensemble des populations enregistrées qui lui succède. Afin de mettre en valeur l'évolution de la répartition des DAG au fil des générations, nous avons appliqué la KPCA aux individus de deux populations enregistrées consécutivement : nous comparons les populations obtenues aux générations 2 000 versus 4 000, puis 4 000 versus 6 000, et ainsi de suite. On notera que les échelles apparaissant sur les différentes figures ne sont pas comparables d'une figure à l'autre. Il s'agit d'un représentation qualitative d'une population de graphes. Pour étudier la dispersion ou la concentration des graphes au fil des générations, on comparera les deux populations représentées au sein d'une même figure. Enfin, nous représentons l'évolution de la population dans des algorithmes évolutionnaires divergeant au niveau de l'application du niching, à la fois pour la recombinaison parentale et la recombinaison relationnelle. Cette fois cependant, nous avons considéré un taux d'échangé élevé $(0, 4)$ pour les deux méthodes de recombinaison. Pour éviter que la mutation interfère avec le niching quant au contrôle de l'homogénéisation, elle a été désactivé pour cette tranche de tests. Pour réaliser ces tests, nous avons utilisé la *SVM and Kernel Methods Matlab Toolbox* [Can].

La première et la seconde colonne de la figure 6.4 représentent l'évolution de la distribution de la population lorsque la recombinaison relationnelle est utilisée avec ou sans niching respectivement. Ces résultats sont particulièrement intéressants dans la mesure où ils sont corrélés avec la qualité des solutions obtenues par l'AE. En l'absence de niching, la population converge rapidement vers quelques rares points (il ne reste que 4 solutions après 6 000 générations) alors qu'avec le niching, la population conserve de nombreuses solutions distinctes jusqu'à convergence (génération 6 631).

La distribution des solutions d'un AE utilisant la recombinaison parentale est représentée figure 6.5 avec et sans niching dans la première et la seconde colonne respectivement. Dans ce

cas, l'effet du niching est frappant puisque la population de l'AE sans niching converge vers un point unique avant 4 000 générations. Grâce au niching, la population peut préserver une large population hétérogène durant plus de 10 000 générations. Par ailleurs, la population finale semble se structurer en sous-groupes distincts.

À l'issue de ces expériences de visualisation, nous sommes donc en mesure de confirmer certaines conjectures concernant l'influence du niching sur le fonctionnement de notre algorithme évolutionnaire. Le niching retarde l'homogénéisation de la population mais surtout, il préserve une répartition hétérogène des solutions, y compris après que la convergence est atteinte. À ce point de notre étude, la seule question qui demeure quant à nos choix algorithmiques est celle du choix de la méthode de recombinaison. En effet, si la recombinaison relationnelle pourvue d'un taux d'échange élevé a donné les meilleurs résultats dans la plupart des cas, le DC semble réduire considérablement la plupart des différences entre les trois méthodes de recombinaison. La recombinaison parentale notamment présente des performances intéressantes lorsqu'elle est utilisée conjointement avec le *deterministic crowding*. Afin de trancher, l'efficacité des recombinaisons parentale et relationnelle (pourvue d'un fort taux d'échange) ont été étudiées via la représentation de leurs courbes d'apprentissage.

6.4.3 Confirmation des résultats par courbes d'apprentissage

Les courbes d'apprentissage que nous présentons à la figure 6.6 rendent compte des résultats obtenus par un algorithme évolutionnaire pour des tailles de bases d'apprentissage croissantes. Les algorithmes ont été testés avec des bases d'apprentissage dont la taille est comprise entre 50 et 400 échantillons avec un pas de 50. Pour chaque taille d'échantillon, les tests ont été répétés 10 fois sur des bases de données distinctes et indépendantes. Les mêmes bases de données ont été utilisées pour tous les algorithmes testés. Les résultats obtenus par chacun d'entre eux sont exprimés en termes de *sensibilité* et de *ppv*. Chaque point de la courbe correspondant à une taille d'échantillon donnée représente la valeur moyenne ainsi que l'écart-type de l'un des indices de qualité sus-cités pour les 10 exécutions mentionnées.
Les courbes d'apprentissage sont fournies pour la recombinaison parentale et la recombinaison relationnelle selon que le *deterministic crowding* est ou non utilisé. Pour les deux méthodes de recombinaison, nous nous sommes bornés à considérer un taux d'échange élevé dans la mesure où celui-ci apparaît clairement favorable à une bonne exploration de l'espace de recherche dans nos résultats préliminaires.
On peut constater que pour la *ppv* (figure 6.6 [A1, A2]) et la *sensibilité* (figure 6.6 [B1, B2]), la recombinaison relationelle donne de meilleurs résultats que la recombinaison parentale, même si la différence entre ces deux méthodes est plus frappante en l'absence de crowding (figure 6.6 [A1, B1]) qu'avec crowding (figure 6.6 [A2, B2]). Cela tend à confirmer nos premières observations. Les résultats précédents, obtenus pour une base d'apprentissage de 300 individus, mettaient en avant la supériorité de la recombinaison relationnelle. Il s'avère que cela demeure vrai pour des tailles d'échantillons variables.

Au vu de ces résultats, nous avons choisi d'utiliser la recombinaison relationelle avec un taux d'échange élevé conjointement avec la mutation et le Deterministic Crowding pour mener les tests suivants.

6.5 Comparaison avec des algorithmes d'apprentissage alternatifs

À l'issue des expériences précédentes, nous avons retenu un algorithme génétique reposant sur une recombinaison relationnelle avec un taux de recombinaison élevé et utilisant conjointement la mutation et le *deterministic crowding*. Pour les tests que nous allons présenter, la taille de la population (200) ainsi que le critère d'arrêt de l'algorithme restent inchangés. Afin de valider cet algorithme génétique, nous l'avons comparé à des méthodes fréquemment utilisées dans la littérature pour l'apprentissage de structures dans les réseaux Bayésiens.
Les performances de ces méthodes ont été étudiées en s'appuyant sur les courbes d'apprentissage rendant compte de la *ppv* et de la *sensibilité* des solutions générées. Ces courbes ont été construites selon le procédé décrit dans la section précédente. Des tailles d'échantillons croissantes sont prises en compte. Pour chacune d'entre elles, chaque algorithme a été confronté à 10 jeux de données indépendants. Les moyennes et les écarts-types des résultats obtenus à l'issu de chaque exécution sont représentés sur la courbe pour la taille d'échantillon correspondante. Comme auparavant, les indices de qualité ont été calculés à partir des CPDAG correspondant aux solutions rendues par ces algorithmes.

6.5.1 Les algorithmes d'apprentissage utilisés pour la comparaison

Méthodes d'exploration de l'espace des DAG Tout d'abord, nous avons considéré des méthodes classiques d'exploration de l'espace des DAG : les algorithmes K2 et de montée de colline. Comme nous l'avons expliqué à la section 4.3.3.2 page 104, l'algorithme de montée de colline est une procédure de recherche gloutonne qui se déplace à chaque itération vers le DAG maximisant le score BIC au sein de son voisinage. Ce voisinage est généré en considérant les additions, les suppressions ainsi que les inversions de chacun des arcs du DAG courant. Tout comme l'AE, l'algorithme de montée de colline est initialisé aléatoirement.
L'algorithme K2 présenté à la section 4.3.3.2 page 105 est également un algorithme glouton, mais il opère au sein d'un espace dans lequel tous les DAG doivent respecter un ordre topologique. Compte tenu de la grande sensibilité de cet algorithme à l'initialisation [FL04], il nous a semblé inutile de le tester en lui soumettant un ordre topologique aléatoire. Nous avons donc choisi de lui fournir le véritable ordre topologique des sommets, tel qu'il apparaît dans le modèle de référence. Il s'agit là d'une connaissance *a priori* très importante qui donne à cet algorithme un avantage conséquent sur ses concurrents. Fondamentalement, la comparaison entre K2 et les autres algorithmes n'est pas « juste », K2 étant trop favorisé par rapport à ses concurrents. Le but ici est justement de créer artificiellement un super-algorithme faisant office de « méthode à abattre » pour les algorithmes éviolutionnaires. Le moyen le plus simple pour cela est de fournir une information a priori conséquente à un algorithme efficace. Il se trouve que cela est particulièrement naturel pour K2.

Méthodes d'échantillonnage de l'espace des DAG Nous avons également utilisé la méthode MCMC [FK03, LBU04] (voir section 4.3.3.2 page 107). Au lieu de chercher à identifier une unique solution maximisant le BIC, MCMC échantillonne la distribution *a posteriori* des structures de réseaux Bayésiens $P(S \mid D)$. Comme nous l'avons évoqué en page 96, cette dernière peut être approchée par le BIC [NRF04]. En partant d'un DAG généré aléatoirement, MCMC construit une chaîne de Markov suivant l'algorithme de Metropolis-Hastings. Après un nombre d'itérations suffisamment grand, un état de cette chaîne (correspondant ici à un DAG) peut être considéré comme un échantillon de $P(S \mid D)$. À chaque exécution de l'algorithme, nous avons effectué

44 000 itérations. Les 40 000 premières constituent » la préchauffe « permettant à la chaîne de Markov d'atteindre son état stationnaire. Les 4 000 suivantes constituent l'échantillon que nous avons conservé. Afin de pouvoir utiliser les même indices de qualité (*ppv* et *sensibilité*) que pour les autres méthodes, un graphe consensus est extrait de cet échantillon. Comme dans l'article de Le et collègues [LBU04], nous n'introduisons dans ce graphe consensus que les arcs dont la fréquence au sein de l'échantillon retenu dépasse $0,5$. Ainsi, nous ne conservons que les arcs présents dans la majorité des CPDAG correspondants aux DAG échantillonnés. Ces arcs représentent les interactions présentes au sein des solutions les plus fréquemment échantillonnées et par conséquent, ils appartiennent plus vraisemblablement au véritable modèle.

Il faut noter qu'au même titre que l'AE, les trois algorithmes précédents ont été utilisés de telle sorte que les sommets de toute solution candidate ne puissent avoir plus de 10 parents. De nouveau, outre une réduction bienvenue de l'espace de recherche, cette contrainte permet surtout de limiter les temps de calcul du BIC qui devient rédhibitoire lorsque les tables de probabilités conditionnelles sont trop grandes.

Méthodes d'apprentissage par contraintes Pour finir, nous avons également testé des méthodes d'apprentissage par contraintes. Plus précisément, nous avons considéré des méthodes hybrides introduites en section 4.3.4 page 108 : l'algorithme MMHC [TBA06] et l'algorithme BN-PC-B [CGK$^+$02]. Ces deux approches construisent la structure d'un réseau Bayésien en testant les indépendances conditionnelles entre les attributs des données. Pour cela, nous avons choisi d'utiliser le test du χ^2, avec un seuil de confiance de $0,05$. Rappelons que BN-PC-B construit d'abord un arbre orienté en utilisant la méthode de l'arbre de recouvrement maximum de Chow et Liu [CL68]. Une série de tests d'indépendances conditionnelles est ensuite réalisée pour ajouter des arêtes manquantes à cet arbre, éliminer les arêtes surnuméraires et détecter les V-structures. On obtient alors un CPDAG, correspondant à une classe d'équivalence.
L'algorithme MMHC construit quant à lui le squelette du réseau Bayésien en suivant la même procédure que l'algorithme PC avant d'utiliser une méthode de montée de colline pour orienter les arcs. Classiquement, cette dernière étape est réalisée de telle sorte que l'on cherche à maximiser le score BD [HG95].

Les logiciels utilisés Pour réaliser ces tests, nous avons utilisé les implantations de K2 et de MCMC disponibles dans *BNT* [Mur]. L'algorithme de montée de colline provient du paquet *BNT Structure Learning Package* [Ler]. Enfin, les algorithmes MMHC et BN-PC-B testés sont disponibles dans le logiciel *Causal Explorer* [ATSB03].

6.5.2 Résultats

Tout d'abord nous considérons les résultats des méthodes hybrides. L'algorithme BN-PC-B présente les plus mauvais résultats, que ce soit en termes de *sensibilité* ou de *ppv*. Son initialisation, réalisée avec la méthode de l'arbre de recouvrement maximum, produit un arbre orienté qui ne permet pas d'approcher précisément la structure objectif. On peut supposer que les tests d'indépendances conditionnelles qui s'en suivent ne parviennent pas à identifier les arcs manquants ou surnuméraires, probablement du fait de la faible quantité de données utilisées pour réaliser ces tests. L'algorithme MMHC, qui est fondé sur le même principe de recherche des indépendances conditionnelles, a également une *sensibilité* médiocre. Il présente par contre une *ppv* relativement élevée. Manifestement, la seconde phase de cet algorithme (qui repose sur une recherche gloutonne) permet de produire nettement moins de faux positifs, notamment en assurant une meilleure orientation des arcs identifiés. Bien sûr, un meilleur réglage du

seuil de confiance utilisé pour les tests d'indépendances conditionnelles devrait nous permettre d'améliorer ces résultats. Par exemple, un seuil de confiance plus faible devrait nous permettre d'éliminer un certain nombre de faux positifs, mais au prix d'une perte de *sensibilité*. Toutefois, avec le paramètrage que nous avons proposé, ces deux algorithmes sont dépassés par les méthodes d'exploration de l'espace des DAG (exception faite de MMHC pour la *ppv*).

Parmi les différentes heuristiques de recherche, l'algorithme de montée de colline donne les plus mauvais résultats, que ce soit en termes de *sensibilité* ou de *ppv*. En effet, il s'agit d'une heuristique déterministe qui converge vers le maxima local le plus proche de la solution que nous générons aléatoirement à l'initialisation. L'algorithme MCMC, qui repose sur une heuristique stochastique donne de meilleurs résultats. Toutefois, si l'on considère la variabilité de ces méthodes, nous notons que les performances de la recherche gloutonne et de l'algorithme MCMC sont relativement proches.

En revanche, l'algorithme génétique, qui surpasse légèrement MCMC en termes de *sensibilité*, donne des résultats très supérieurs à la recherche gloutonne et à MCMC en termes de *ppv*. Il est vrai que les courbes de performance de l'algorithme MCMC dépendent de la valeur du seuil à partir duquel on considère que la fréquence d'un arc parmi les 4 000 DAG générés est suffisante pour l'inclure dans le graphe consensus. Ce paramètre que nous avons ici fixé arbitrairement à 50% rend plus difficile la comparaison de MCMC avec d'autres méthodes. L'utilisation de courbes ROC, visant à trouver le seuil donnant le meilleur compromis entre *sensibilité* et *ppv* est ici difficile à mettre en œuvre. En effet, les résultats obtenus d'une base d'apprentissage à l'autre présentent trop de variabilité pour qu'un paramètre de seuil optimal puisse émerger. Toutefois, le large écart entre les courbes de *ppv* de l'algorithme génétique et du MCMC semble garantir la prédominance du premier sur le second.

Finalement, l'algorithme K2 est le seul qui surpasse l'AG en termes de *ppv*. Cela était prévisible car K2 génère naturellement moins de faux positifs que les autres méthodes. En effet, l'information *a priori* est particulièrement précieuse lorsque l'on apprend à partir d'un petit nombre d'exemples et grâce à l'ordre topologique que nous lui avons fourni, K2 réalise la recherche dans un espace des solutions plus petit comprenant un nombre d'arcs restreint. Malgré cela, les performances de K2 concernant la *sensibilité* sont disputées par l'AG pour des tailles d'échantillons supérieures à 200.

Comme l'ont montré Leray et Francois [FL04], utiliser un arbre généré par la méthode de Chow et Liu pour initialiser l'algorithme de montée de colline aurait permis d'améliorer significativement les solutions générées par ce dernier. Cependant, pour être juste, il faudrait alors proposer des méthodes d'initialisation similaires pour les autres heuristiques. Cela devient problématique pour l'algorithme évolutionnaire qui est initialisé avec une population de solutions. Il est envisageable d'utiliser l'algorithme MCMC pour générer un échantillon de solutions prometteuses à soumettre à l'algorithme évolutionnaire. La difficulté qui apparaît alors est d'ordre purement calculatoire, les temps de calcul cumulés de ces deux algorithmes rendant cette approche difficilement exploitable dans un cadre expérimental (lorsqu'un grand nombre de tests sont requis).

Pour finir, précisons que le nombre d'évaluations de la fonction objectif (le score BIC) réalisé par l'algorithme MCMC, l'algorithme de montée de colline et l'algorithme génétique, est de l'ordre de quelques dizaines de milliers pour toutes ces méthodes. Dans la mesure où le calcul de la fonction objectif accapare l'essentiel du temps de calcul d'un algorithme de recherche,

nous pouvons donc dire que ces trois algorithmes ont des coûts de calcul du même ordre. Leur comparaison paraît donc fondée. Les autres algorithmes étant déterministes, leur coût de calcul importe peu. En effet, même si K2 ou MMHC sont nettement moins coûteux que les algorithmes que nous venons de citer, il n'est pas possible de recourir à un procédure d'initialisation multiple.

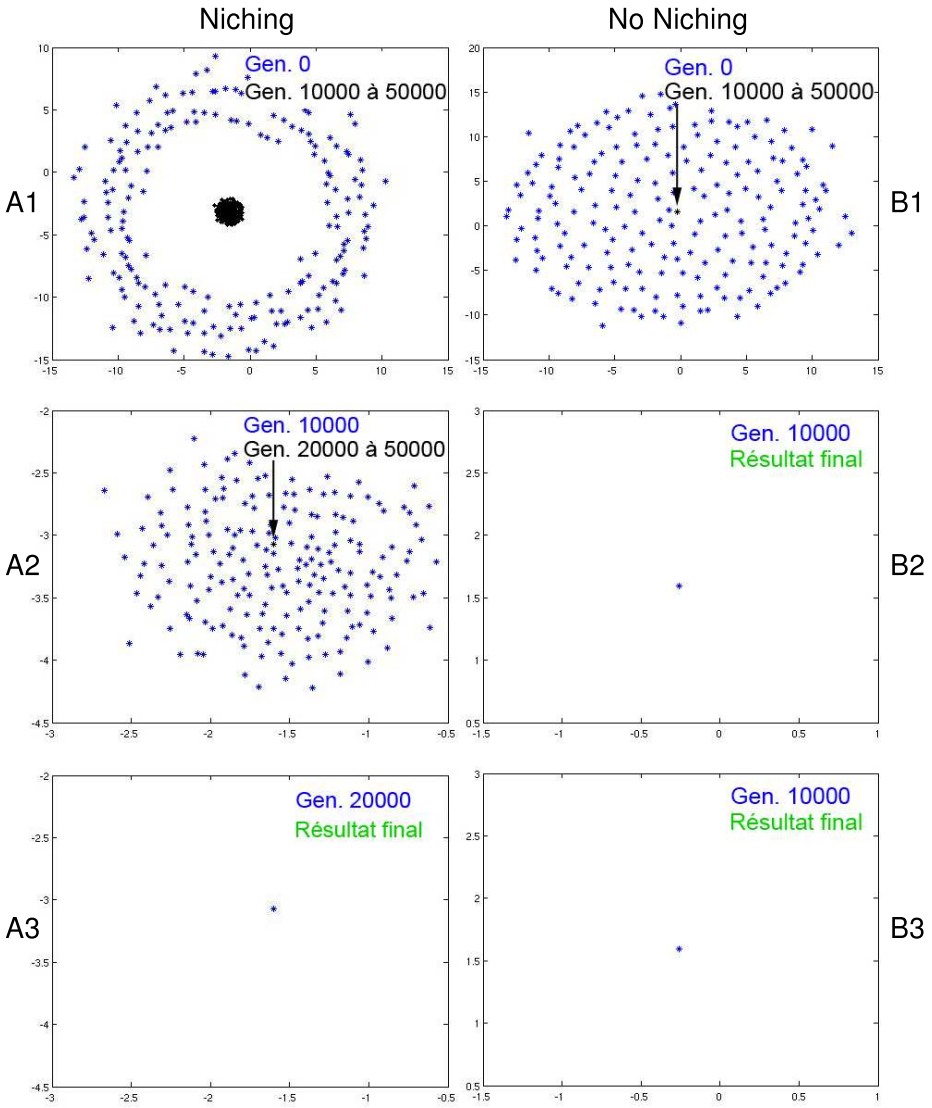


FIG. 6.2 – Représentation des populations successives d'un AE utilisant la recombinaison relationnelle par Sammon-mapping. Ces figures montrent l'évolution de la distribution de la population au cours d'un AE utilisant la recombinaison **relationnelle** (avec un taux d'échange de $0,4$) avec niching (A1-A3) et sans niching (B1-B3). Les populations de DAG ont été enregistrées toutes les 10 000 générations jusqu'à la génération 50 000. Chaque figure compare la répartition des DAG appartenant à une population donnée à celle des DAG issus des populations suivantes. Les DAG de ces populations sont représentés sous la forme de points sur une carte 2D grâce à l'utilisation du *Sammon-mapping*.

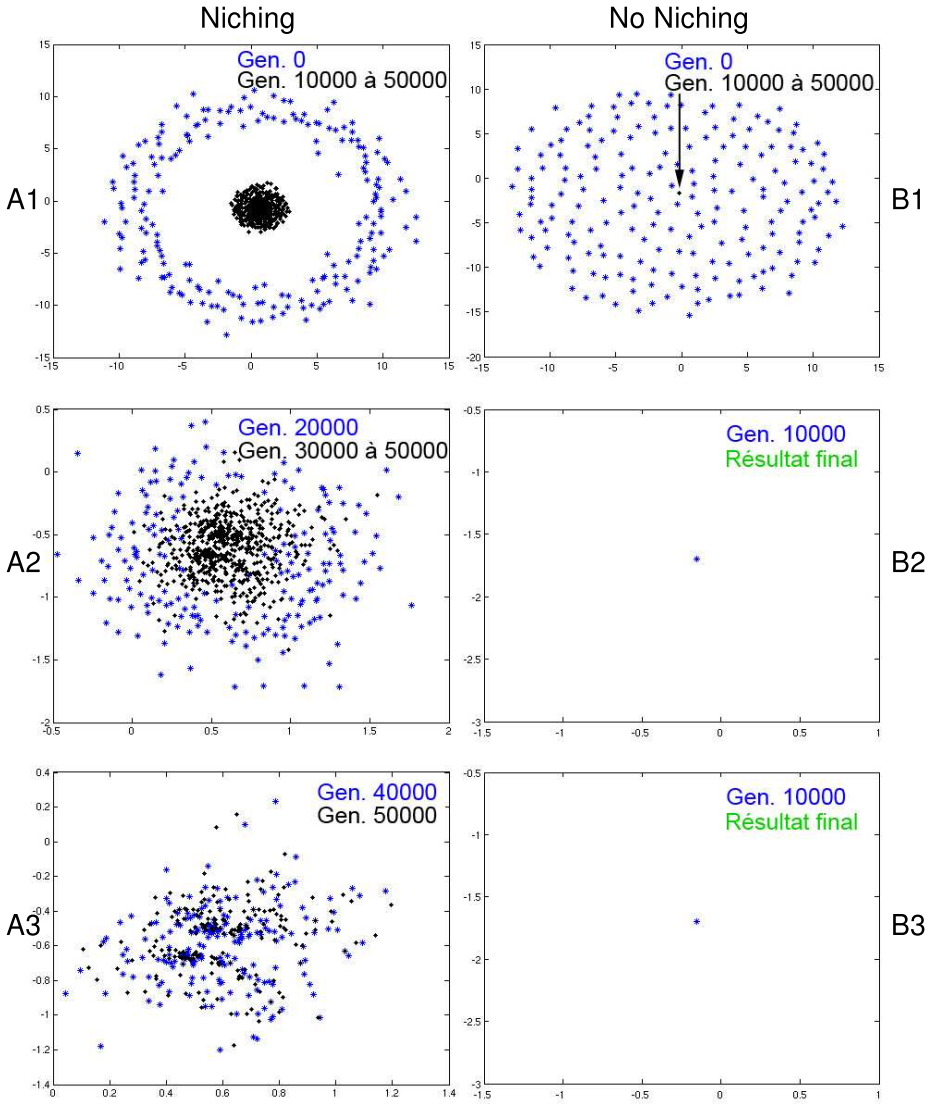


FIG. 6.3 – Représentation des populations successives d'un AE utilisant la recombinaison parentale par Sammon-mapping. Ces figures montrent l'évolution de la distribution de la population au cours d'un AE utilisant la recombinaison **parentale** (avec un taux d'échange de $0,1$) avec niching (A1-A3) et sans niching (B1-B3). Les populations de DAG ont été enregistrées toutes les 10 000 générations jusqu'à la génération 50 000. Chaque figure compare la répartition des DAG appartenant à une population donnée à celle des DAG issus des populations suivantes. Les DAG de ces populations sont représentés sous la forme de points sur une carte 2D grâce à l'utilisation du *Sammon-mapping*.

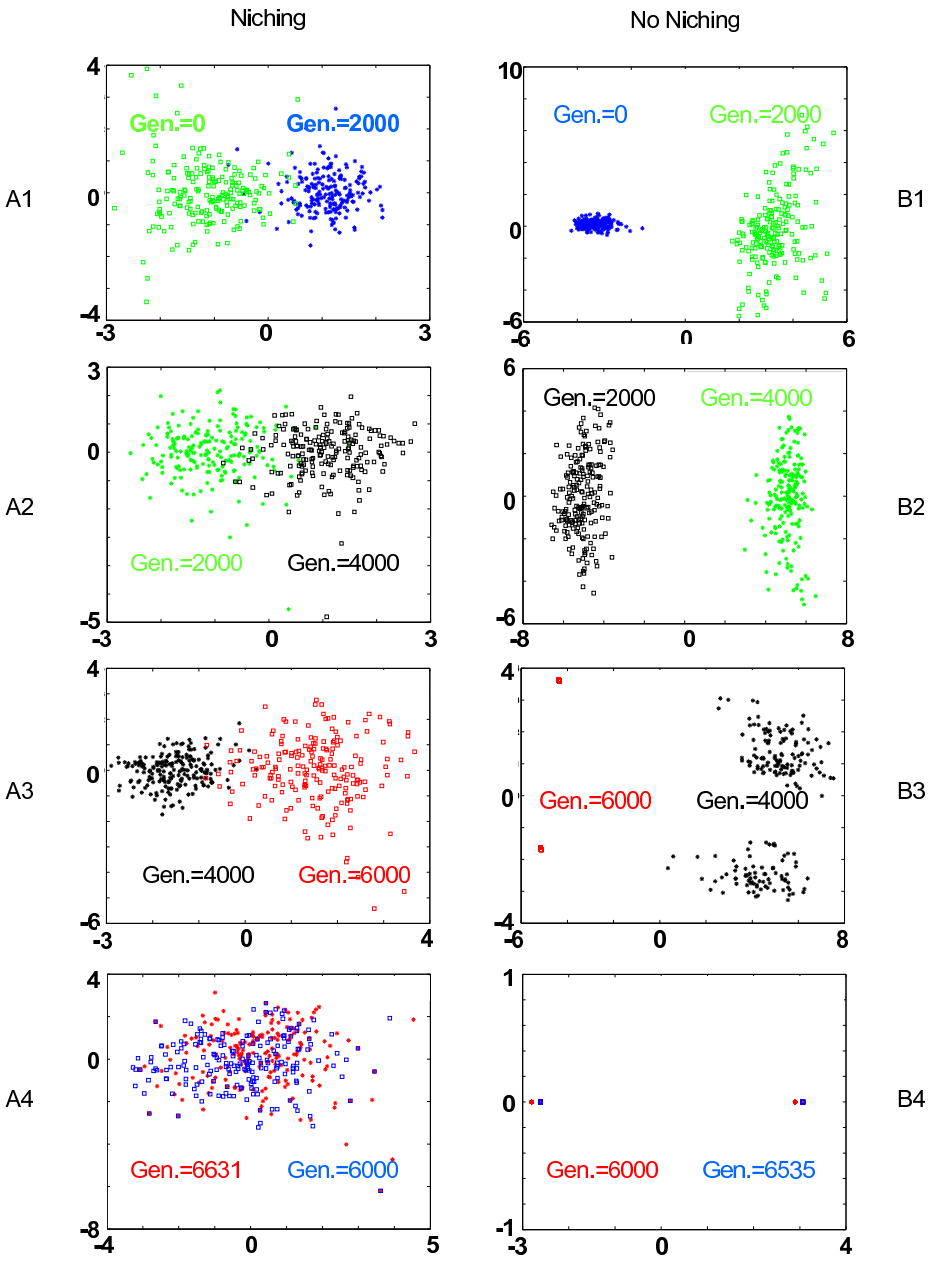


FIG. 6.4 – Représentation des populations successives d'un AE utilisant la recombinaison relationnelle par KPCA. Ces figures montrent l'évolution de la distribution de la population au cours d'un AE utilisant la recombinaison **relationnelle** (avec un taux d'échange de $0,4$) avec niching (A1-A4) et sans niching (B1-B4). Les populations de DAG ont été enregistrées toutes les 2 000 générations ainsi qu'après convergence de l'algorithme. Chaque figure représente les graphes issus de deux populations enregistrées consécutivement avec un pas de 2 000. Les graphes de ces deux populations sont représentés sous la forme de points sur une carte 2D grâce à l'utilisation de l'*Analyse en Composantes Principales Kernelisée*.

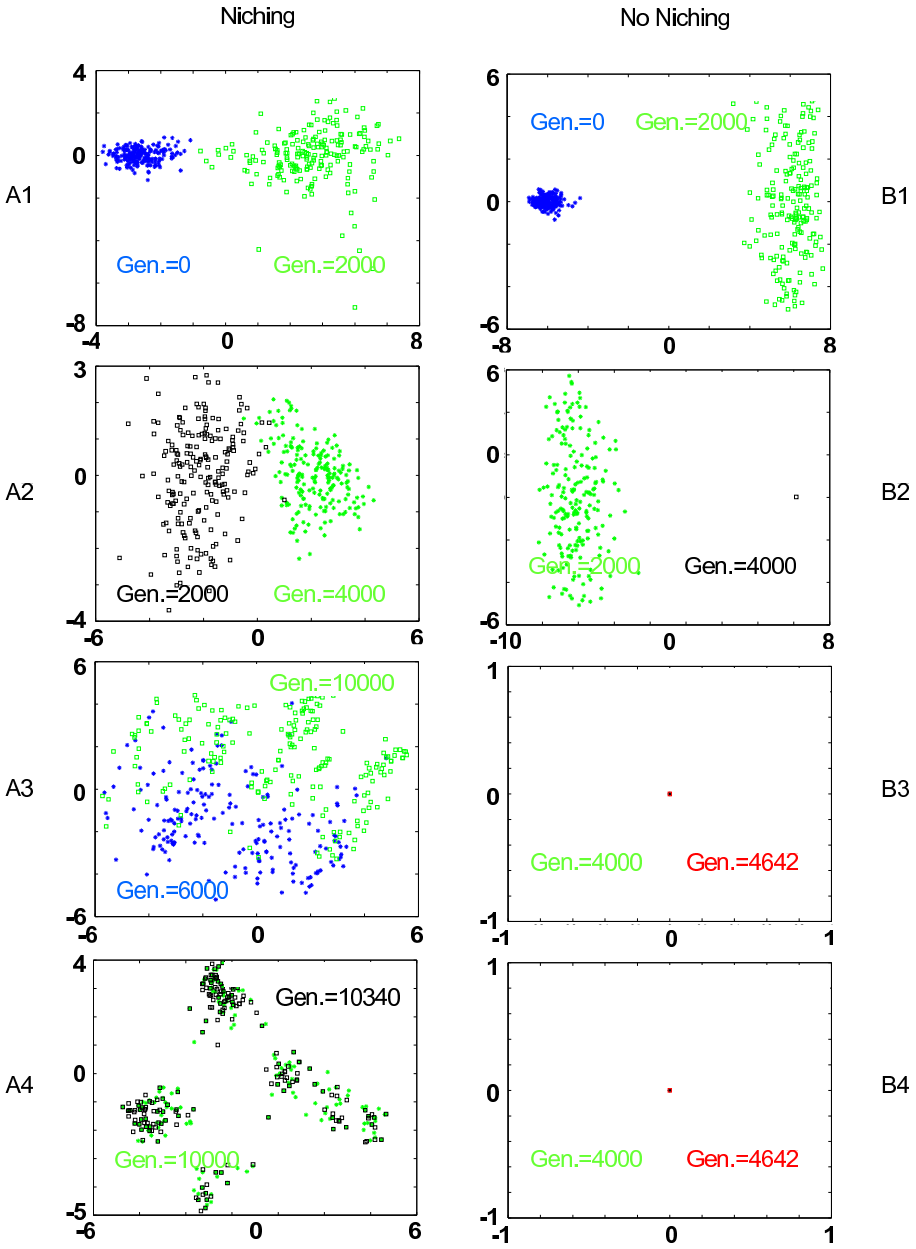


FIG. 6.5 – Représentation des populations successives d'un AE utilisant la recombinaison parentale par KPCA. Ces figures montrent l'évolution de la distribution de la population au cours d'un AE utilisant la recombinaison **parentale** (avec un taux d'échange de $0,4$) avec niching (A1-A4) et sans niching (B1-B4). Les populations de DAG ont été enregistrées toutes les 2 000 générations ainsi qu'après convergence de l'algorithme. Chaque figure représente les graphes issus de deux populations enregistrées consécutivement avec un pas de 2 000. Les graphes de ces deux populations sont représentés sous la forme de points sur une carte 2D grâce à l'utilisation de l'*Analyse en Composantes Principales Kernelisée*.

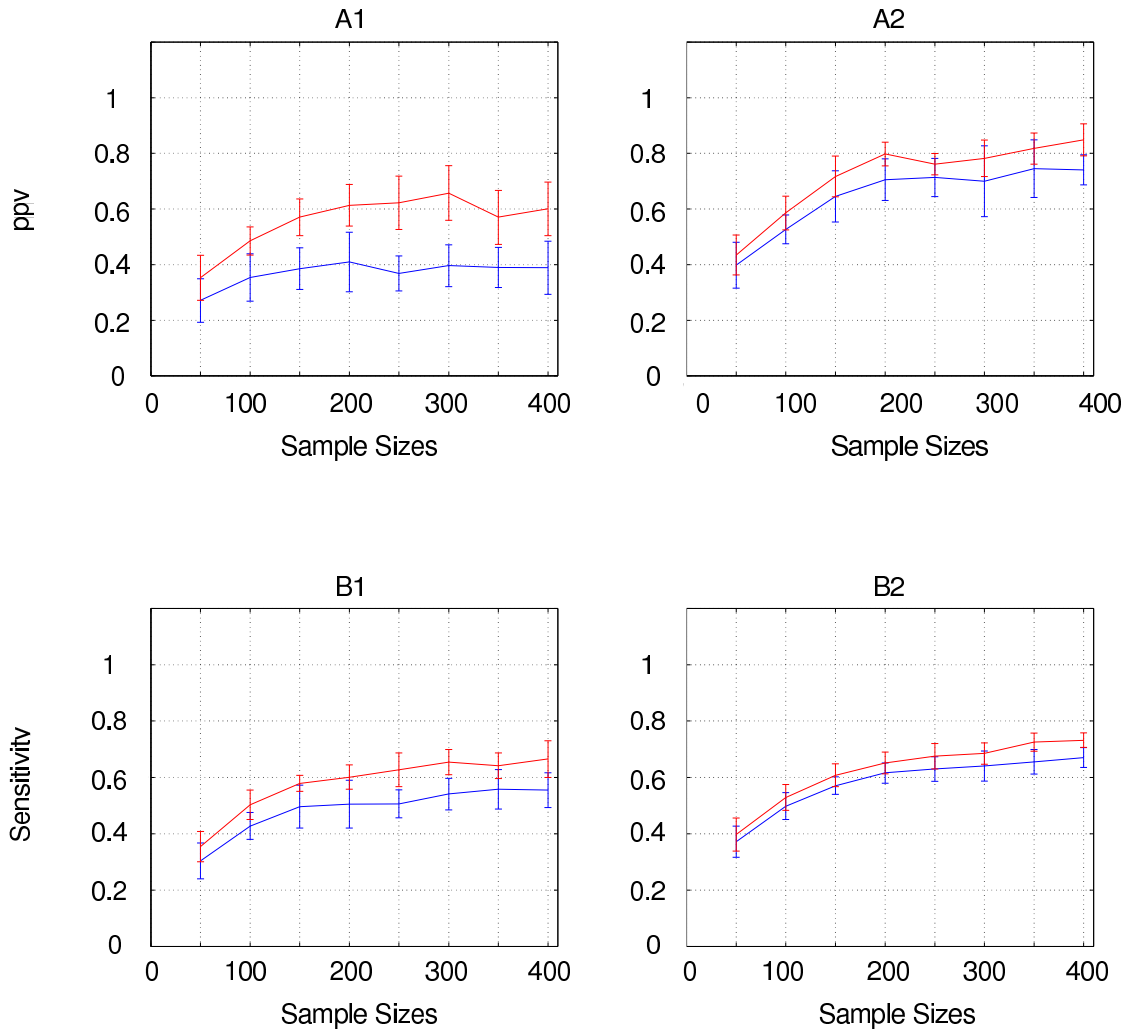


FIG. 6.6 – Comparaison des courbes d'apprentissage obtenues avec les recombinaisons parentale et relationnelle. Pour chaque algorithme d'apprentissage, les résultats de la comparaison entre graphe appris et graphe de référence sont exprimés en termes de *valeur de prédiction positive* (A1 et A2) et de *sensibilité* (B1 et B2). Les sous-figures A1 et B1 montrent les résultats obtenus sans niching, alors que les sous-figures A2 et B2 montrent les résultats obtenus avec niching. Le codage des couleurs est **bleu** pour la recombinaison parentale et **rouge** pour la recombinaison par lien. Pour chaque taille d'échantillon, les tests sont répétés sur 10 bases d'apprentissages distinctes et indépendantes. Les mêmes jeux de données sont utilisés pour tous les AE. Chaque point sur les courbes correspond à une taille d'échantillon donnée et représente la valeur moyenne ainsi que l'écart-type de la mesure de qualité considérée sur les 10 exécutions de l'algorithme.

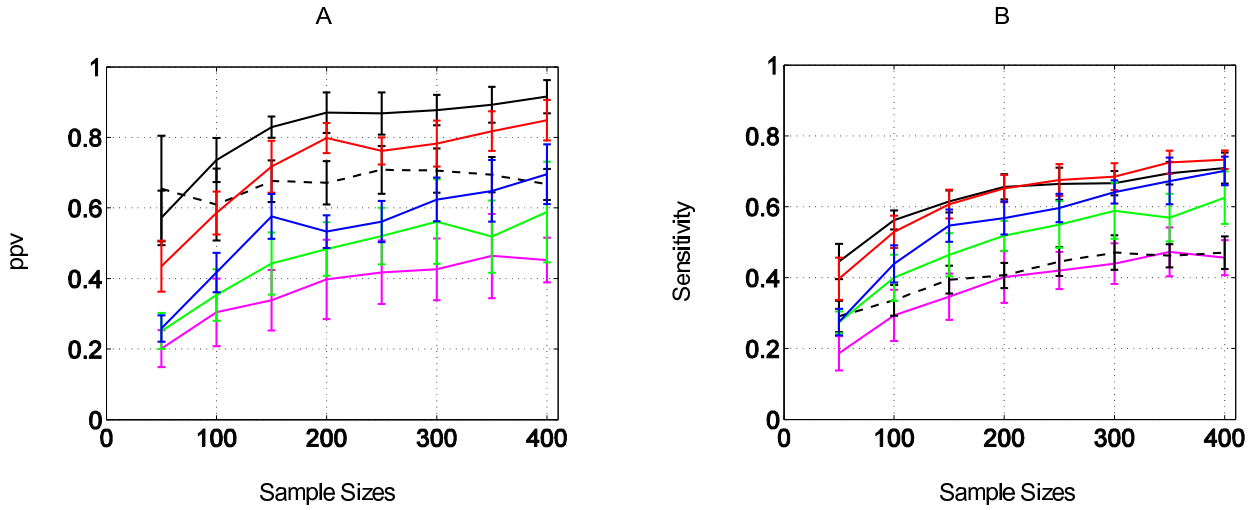


FIG. 6.7 – Comparaison des courbes d'apprentissage pour six méthodes d'apprentissages distinctes. Pour chaque algorithme d'apprentissage, les résultats de la comparaison entre graphe appris et graphe de référence sont exprimés en termes de *valeur de prédiction positive* (A) et de *sensibilité* (B). Le codage des couleurs est *magenta* pour BN-PC, *vert* pour la recherche gloutonne, *bleu* pour l'algorithme MCMC, *noir* pour l'algorithme K2, *pointillé-noir* pour MMHC et *rouge* pour l'algorithme génétique. Pour chaque taille d'échantillon, les tests sont répétés sur 10 bases d'apprentissages distinctes et indépendantes. Les mêmes jeux de données sont utilisés pour tous les algorithmes d'apprentissage. Chaque point sur les courbes correspond à une taille d'échantillon et représente la valeur moyenne ainsi que l'écart-type de la mesure de qualité considérée sur les 10 exécutions de l'algorithme.

Chapitre 7

Des algorithmes génétiques aux algorithmes à estimation de distribution : les perspectives

Dans ce chapitre, nous présentons une nouvelle classe d'algorithmes évolutionnaires qui s'est fortement développée au cours des années 90 : les algorithmes à estimation de distribution (*Estimation of Distribution Algorithms* ou EDA [MP96]), également appelés algorithmes génétiques à construction de modèle probabiliste (*Probabilistic Model-Building Genetic Algorithms* ou PMBGA). Les EDA sont une famille de méthodes reposant sur une conception originale des opérateurs de variation. On propose de remplacer la recombinaison ainsi que la mutation, qui assurent une ré-association et une modification explicites des paramètres caractérisant les solutions candidates, par une approche probabiliste. L'idée est d'utiliser l'information globale contenue dans un ensemble de solutions prometteuses à la génération g pour estimer leur distribution. Les nouvelles solutions candidates constituant la population à la génération $g+1$ sont alors échantillonnées à partir de cette distribution. Les EDA utilisent donc des techniques d'apprentissage automatique pour résoudre des problèmes d'optimisation en essayant d'apprendre la localisation des régions les plus prometteuses de l'espace de recherche. Un modèle probabiliste est utilisé afin de générer des solutions candidates et l'apprentissage permet d'estimer ce modèle probabiliste à chaque génération. Nous proposons ici une étude exploratoire, visant à démontrer la pertinence des EDA pour les tâches d'optimisation qui nous intéressent. Les résultats qui sont fournis à la fin de ce chapitre ne doivent donc pas être perçus comme définitifs, mais comme un moyen de nous ouvrir de nouvelles perspectives pour des travaux futurs.

Tout d'abord, nous rappelons quelques concepts et notations utiles à la bonne compréhension de cette famille de méthodes. Pour assurer la généricité de cette présentation nous considérons le problème d'optimisation binaire :

$$\max f(\mathbf{G})$$

où $\mathbf{G} = (G_1, G_2, \ldots, G_n) \in \Omega = \{0,1\}^n$ est une solution candidate appartenant à un espace binaire de dimension n et $f : \Omega \to \mathbb{R}^+$ la fonction objectif. La plupart des algorithmes présentés ci-dessous peuvent être aisément étendus à des problèmes d'optimisation discrète. Chaque élément G_i d'un vecteur G appartient alors à un alphabet de taille finie.

Les EDA, au même titre que les autres algorithmes évolutionnaires, maintiennent et améliorent une population de solution candidates de taille finie, notée M, au fil des générations. Soit $Pop(t)$ la population à la génération t. À $t = 0$, l'algorithme est initialisé par la génération aléatoire de M solutions appartenant à Ω. Tous les EDA obéissent alors au processus itératif suivant :

Sélection pour la reproduction Sélection d'un ensemble de solutions prometteuses (maximisant une fonction de performance) à partir de $Pop(t)$ pour former un ensemble de parents $PopPar(t)$.
Des méthodes classiques de sélection conviennent pour cette étape.

Reproduction

Apprentissage du modèle probabiliste Construction ou mise à jour d'un modèle permettant d'estimer la distribution des solutions dans $PopPar(t)$. On se ramène à un problème d'apprentissage où, étant donnée une famille de modèles $\mathcal{H}$, on essaie d'identifier un modèle $h \in \mathcal{H}$ décrivant au mieux la distribution des solutions candidates dans $PopPar(t)$. Lorsque la structure du modèle n'est pas fixée, il faut apprendre à la fois sa structure et ses paramètres. À la génération t, le modèle estimé est noté h_t.

Échantillonnage du modèle probabiliste Génération aléatoire de nouvelles solutions candidates appartenant à l'espace de recherche Ω par échantillonnage de $h_t(\mathbf{G})$. Ces solutions enfants sont contenues dans $PopEnf(t)$.

Sélection pour le remplacement On procède au remplacement total ou partiel des éléments de $Pop(t)$ par les solutions de $PopEnf(t)$ afin de créer une nouvelle population de taille M : $Pop(t+1)$.
Là encore, des méthodes classiques de sélection peuvent être utilisées.

Notons qu'à l'exception de la phase d'apprentissage interne à ces algorithmes, le reste des étapes d'un EDA est similaire à celles mises en œuvre dans un algorithme génétique. Par conséquent, les notations et les termes utilisés dans le chapitre 5 peuvent être repris sans encombre.

Le principal attrait des EDA réside dans leur capacité à apprendre la structure du problème d'optimisation. Les algorithmes génétiques nécessitent une connaissance approfondie du problème pour choisir un codage et une méthode de recombinaison permettant d'identifier et d'extraire les *briques de base* (BB) des solutions optimales. Le but de la phase de codage est généralement de mettre en évidence au sein des chromosomes des sous-chaînes contenant les variables devant être optimisées de concert, c'est-à-dire celles qui présentent de fortes interactions. L'utilisation d'un modèle probabiliste permet théoriquement de capturer ces interactions entre gènes virtuels à partir de la base d'exemples que constitue la population des parents. Par conséquent, le choix de $\mathcal{H}$ est critique dans la conception d'un EDA dans la mesure où la famille de modèles utilisée doit permettre de capturer les dépendances entre gènes virtuels au sein de la population. L'utilisation de modèles probabilistes suffisamment riches semble donc s'imposer pour traiter des problèmes d'optimisation complexes. Paradoxalement, il est nécessaire de s'assurer que $h(x)$ soit suffisamment simple pour pouvoir être estimée et échantillonnée efficacement. En effet, le risque est d'utiliser un modèle susceptible d'approcher précisément la distribution des solutions dans $PopPar(t)$ mais trop difficile à estimer dans les faits. La solution résultante peut alors est inférieure à celles obtenues avec des modèles plus simples. Un point important dans les EDA est que l'apprentissage durant la phase de reproduction ne souffre théoriquement pas du manque de données. En effet, la taille de la base d'apprentissage (qui est fonction de M, la taille de la population globale) est fixée par l'utilisateur. Il est donc possible d'échantillonner autant d'individus qu'on le souhaite à partir du modèle h_t pour générer $Pop(t+1)$. Dans la pratique, cet avantage est difficilement exploitable. En effet, il est nécessaire d'évaluer la performance de chacune des solutions enfants préalablement aux étapes de sélection (remplacement et reproduction). Dans la mesure où il s'agit de la phase la plus lourde de l'algorithme en termes de calculs, augmenter la taille de la population implique une augmentation conséquente des temps de calcul.

Le choix d'un bon modèle de distribution pour les solutions candidates, respectant un équilibre entre la capacité à capturer des dépendances complexes entre gènes virtuels et à être appris efficacement, est donc crucial. Pour l'essentiel, les EDA divergent donc par la classe de modèles choisie pour représenter la distribution d'un ensemble de solutions pré-sélectionnées ainsi que par les méthodes permettant d'apprendre puis d'échantillonner un tel modèle. Nous présentons par la suite une sélection d'EDA mettant en oeuvre des familles de distribution de probabilité de complexité croissante. Cette sélection n'est pas exhaustive, de nombreux raffinements ayant été introduits au cours du temps au sein de ces différentes approches. Elle permet néanmoins de capturer l'esprit et les enjeux de cette famille d'algorithmes.

7.1 Les modèles sans dépendances

Le moyen le plus simple d'estimer une distribution est de supposer que les variables sont indépendantes. Dans ce cas, le modèle de distribution des solutions de $PopPar(t)$ est simplement constitué par un ensemble de distributions marginales. Ces dernières sont estimées par les fréquences d'apparition des différentes valeurs prises par un gène virtuel (ici 0 ou 1) dans la population, et ce pour chacun des gènes virtuels. Ces fréquences sont utilisées pour guider la recherche de solutions optimales via la génération de nouveaux chromosomes. Ces derniers sont construits en tirant une valeur pour chaque gène virtuel dans la distribution marginale correspondante. De la sorte, les BB d'ordre 1 sont reproduites et échangées très efficacement. Les algorithmes fondés sur ce principe fonctionnent très bien sur des problèmes linéaires où les variables ne présentent pas d'interactions entre elles [Müh97, HLG99].

La simplicité du modèle — qui interdit la prise en compte de toute interaction entre les gènes virtuels — est alors compensée par la simplicité des calculs menant à son estimation. Dans cette catégorie d'EDA, on peut notamment citer les algorithmes à distribution marginale uni-variée (*Univariate Marginal Distribution Algorithms* ou UMDA [MP96]), l'apprentissage incrémental à base de population (*Population Based Incremental Learning* ou PBIL [Bal94]) ou l'algorithme génétique compact (*Compact genetic algorithm* ou cGA [HLG99]) où toutes les variables de $P(\mathbf{G})$ sont indépendantes, i.e.

$$P(\mathbf{G}) = P(G_1)P(G_2)\cdots P(G_n)$$

où $P(G_i)$ est la probabilité marginale de la variable G_i. On notera que celle-ci peut être estimée par un simple calcul de fréquence comme c'est le cas pour l'UMDA qui est l'EDA le plus simple. Par ailleurs, une solution $\mathbf{g^j}$ peut être aisément construite par un tirage $\mathbf{g^j} = (g_1^j, g_2^j, \ldots, g_n^j)$ dans la distribution présentée ci-dessus, chaque g_i^j pouvant être généré indépendamment des autres à partir de $P(G_i)$. À la génération t, on construit ainsi une population de solutions candidates $PopEnf(t) = (\mathbf{g^1}, \mathbf{g^2}, \ldots, \mathbf{g^m})$ où m est le nombre d'enfants générés à l'issue de la reproduction. Comme dans les chapitres précédents, on notera que les variables aléatoires sont représentées par des majuscules alors que les minuscules sont utilisées pour représenter leurs instanciations. Les ensembles de variables aléatoires sont représentés en gras. Enfin, notons que dans le cas binaire $P(G_i)$ est souvent remplacé par $P(g_i = 1)$.

Ces algorithmes donnent de bons résultats pour des fonctions objectif ne présentant pas d'interactions significatives entre leurs composantes, comme c'est notamment le cas pour les fonctions linéaires. Toutefois, pour des fonctions présentant des dépendances réelles entres variables, les algorithmes sus-cités échoueront dans l'identification de l'optimum global.

Dans ce qui suit , nous passons en revue différents algorithmes qui prennent en compte de manières diverses les dépendances entre variables. S'il est vrai que les premiers ne sont pas des

EDA *stricto sensu*, ils en partagent les concepts centraux et constituent une bonne introduction aux premiers algorithmes à estimation de distribution.

7.1.1 Présentation des algorithmes BSC, PBIL et UMDA

Principes communs

1. Problème : optimisation d'une fonction $f : \mathbf{G} \rightarrow \mathbb{R}^+$ avec $\mathbf{G} = \{0,1\}^n$.
2. Population de solutions à la génération t : $Pop(t) = (G^1, \ldots, G^M)$, avec $G^j \in \mathbf{G}$.
3. Chaque solution est générée à partir d'un vecteur de probabilité $P(\mathbf{G}) = (P(G_1), \ldots, P(G_n))$, où $P(G_i)$ représente la probabilité de générer un 1 pour la position i.

$\mathbf{g^j}$	$f(\mathbf{g^j})$	$w = M - rang(\mathbf{g^j})$
$\mathbf{g^1} = 1010$	2	1
$\mathbf{g^2} = 1101$	3	2
$\mathbf{g^3} = 0010$	1	0

Tab. 7.1 – Exemple de l'attribution d'un rang à une solution candidate en fonction de sa performance

Syswerda [Sys89] fut le premier à proposer de remplacer l'opérateur de croisement des algorithmes génétiques par un opérateur plus général, utilisant un vecteur de probabilité $P(\mathbf{G})$. Ces travaux furent motivés par l'observation que les différents types de croisement (un-point, deux-points et uniforme) préservent les proportions de 0 et de 1 pour chaque position G_i au sein d'une population de chaînes binaires. Il a donc proposé de calculer explicitement ces proportions et d'utiliser les probabilités qui en découlent à la place d'un opérateur de croisement, afin de générer de nouveaux individus. Cette approche porte le nom de BSC (*Bit Simulated Crossover*). En outre, Syswerda suggéra de prendre en considération la performance des individus parents afin de biaiser le vecteur $P(\mathbf{G})$ vers des régions intéressantes de l'espace de recherche. Il a donc proposé d'attribuer un poids supérieur aux individus présentant une meilleure performance. Enfin, dans ces travaux, la mutation est considérée comme une simple perturbation de $P(\mathbf{G})$. Elle doit permettre de générer avec une faible probabilité un 0 ou un 1 pour chacune des positions de $\mathbf{G}$.

La première étape de l'algorithme repose sur l'initialisation de toutes les probabilités $P(G_i)$ à $0,5$. Par la suite la règle de mise à jour de cette distribution se traduit de la sorte :

Simuler un croisement Pour chaque bit X_i, $i \in \{1, \ldots, n\}$ on calcule la quantité suivante :

$$P(g_i) \leftarrow \frac{\sum_{\mathbf{g^j} \in Pop(t)} g_i^j . \omega(\mathbf{g^j})}{\sum_{\mathbf{g^j} \in Pop(t)} \omega(\mathbf{g^j})}$$

où $\omega(\mathbf{g^j})$ est le poids associé à l'individu $\mathbf{g^j}$. On notera que si $\omega(\mathbf{g^j}) = 1$ pour tout $\mathbf{g^j} \in Pop(t)$ alors pour tout $i \in \{1, \ldots, n\}$, $P(g_i)$ est égale à la proportion de 1 en G_i dans $Pop(t)$. Pour biaiser $P(g_i)$ vers une valeur de bit intéressante, il est possible de construire le poids $\omega(\mathbf{g^j})$ selon la formule $M - rang(\mathbf{g^j})$, où $rang(\mathbf{g^j})$ est le rang de $\mathbf{g^j}$ au sein de $Pop(t)$ d'après sa fitness $f(\mathbf{g^j})$. Cela signifie que le rang de la meilleure chaîne est 1 et celui de la plus mauvaise est M. Ce propos est illustré figure 7.1.1 où : $P(\mathbf{G}) = \left(\frac{3}{3}, \frac{2}{3}, \frac{1}{3}, \frac{2}{3}\right)$.

Simuler la mutation L'opération suivante est réalisée sur chacun des gènes virtuels G_i, $i \in \{1, \ldots, n\}$:

$$P(g_i) \leftarrow P(g_i)(1 - p_m) + (1 - P(g_i))p_m$$

où p_m est un paramètre de BSC dénotant une probabilité de mutation. Il est possible d'ajuster $P(g_i)$ de telle sorte qu'elle demeure dans l'intervalle $[p_m, 1-p_m]$ afin de maintenir une probabilité minimale p_m d'exploration pour chaque bit.

D'autres algorithmes ont vu le jour dans le sillage de BSC, tel que l'algorithme génétique compact de Harik et collègues [HLG99, Gol89] ou encore l'algorithme PBIL (pour *Population Based Incremental Learning*) développé par Baluja [Bal94]. Ce dernier se distingue notamment de BSC par l'introduction d'une règle de mise à jour du vecteur de probabilité, inspirée de l'apprentissage compétitif tel qu'on le rencontre dans les cartes de Kohonen. Il a également donné lieu à de nombreuses variantes [KPP95, Müh97]. Le fonctionnement de la phase de reproduction l'algorithme PBIL peut être décrit de la manière suivante :

1. Soit $\mathbf{g}^+$ la meilleure chaîne binaire générée au sein de $Pop(t) = (\mathbf{g^1}, \ldots, \mathbf{g^M})$.
2. **Mettre à jour** $P(\mathbf{G})$ de la manière suivante : pour chaque bit en position i,

   $$P(g_i) \leftarrow P(g_i)(1 - LR) + g_i^+.LR$$

   où LR est le taux d'apprentissage (Learning Rate) contrôlant l'amplitude des changements opérés sur $P(\mathbf{g})$ d'une génération sur l'autre.
3. **Effectuer la mutation** : pour chaque bit en position i, modifier $P(g_i)$ avec une probabilité de mutation p_m :

   $$P(g_i) \rightarrow P(g_i) \cdot (1 - mut_{shift}) + mut_{dir} \cdot mut_{shift}$$

   avec :
   - mut_{shift} amplitude de la mutation survenant en chaque position.
   - mut_{dir} renseigne la direction de la mutation, c'est-à-dire 1 ou 0, ces deux alternatives étant équiprobables.

Par conséquent, lorsque $g_i^+ = 0$, $P(g_i)$ diminue proportionnellement à LR.

Les algorithmes que nous venons de voir ne sont cependant pas des EDA dons le sens où ils n'apprennent pas, à chaque génération, un modèle de la distribution des solutions candidates. Dans cette catégorie d'algorithmes, l'EDA le plus simple qui soit est l'algorithme à distribution marginale uni-variée ou UMDA (pour *univariate marginal distribution algorithm*) [MP96]. Ce dernier calcule les fréquences des valeurs prises par chacun des gènes virtuels à partir de l'ensemble de solutions parents. Ces fréquences sont ensuite utilisées pour générer de nouvelles solutions susceptibles de remplacer les anciennes. Le processus est alors répété jusqu'à ce qu'un critère de terminaison soit satisfait. Le plus souvent, cet algorithme se poursuit jusqu'à ce que la valeur de chaque gène virtuel soit fixée. Cela arrive lorsque la probabilité de l'une des valeurs d'un gène virtuel atteint 1 : pour tout $i \in \{1, \ldots, n\}$, $P(g_i = 1) = 1$ ou $P(x_i = 1) = 0$.

7.1.2 Points communs et différences

La principale ressemblance entre ces algorithmes réside dans l'exploitation d'un vecteur de probabilité $P(x)$ ainsi que dans la façon dont ce dernier est utilisé : initialisation de $P(\mathbf{G})$,

génération d'une population $Pop(t)$ à partir de ce dernier, mise à jour de $P(\mathbf{G})$ à partir des individus de $Pop(t)$. Ils divergent cependant par la manière dont la mise à jour de $P(\mathbf{G})$ s'effectue. BSC tend à renouveler $P(\mathbf{G})$ avec toutes les solutions de la population courante, en oubliant tout de la précédente distribution. PBIL propose de rapprocher la distribution $P(x)$ d'une ou de plusieurs solutions prometteuses mais aussi de l'éloigner des solutions de mauvaise qualité. Par ailleurs PBIL introduit un effet mémoire sur les précédentes valeurs de $P(\mathbf{G})$ qui ne sont pas oubliées d'une génération sur l'autre mais conservées et modifiées. Il se démarque fortement des AG classiques par sa capacité à pouvoir n'utiliser qu'une seule solution à chaque génération pour mener l'exploration de l'espace de recherche. Les AG à états stationnaires, même s'il n'exploitent qu'un faible nombre de solutions à chaque génération (typiquement deux) n'en produisent également qu'une ou deux, alors que PBIL, en régénérant une population entière, modifie plus drastiquement l'exploration de l'espace de recherche. *A contrario*, ce qui rapproche PBIL des algorithmes génétiques est l'utilisation de solutions choisies en fonction de leur fitness pour modifier $P(\mathbf{G})$. BSC ne recourt pas explicitement à un processus de sélection, la fitness des individus étant prise en compte directement lors du processus de mise à jour de $P(\mathbf{G})$. Enfin UMDA se situe à mi-chemin entre ces deux méthodes. Il reprend le synopsis classique d'un algorithme génétique, sélectionnant les meilleurs individus afin de mettre à jour $P(\mathbf{G})$. Toutefois cette mise à jour implique de recalculer complètement $P(\mathbf{G})$ sans garder trace des valeurs obtenues à la précédente génération. Bien qu'il soit possible de lui adjoindre un processus de mutation visant à modifier les probabilités de $P(\mathbf{G})$ en ajoutant ou retranchant un bruit gaussien sur tout ou partie des $P(G_i)$, cela n'apparaît pas nécessaire. En effet, la nature même du procédé d'échantillonnage permet de générer un 0 pour une position de bit i même si $P(G_i)$ est élevée. Seul le cas où $P(G_i) = 1$ (ou 0) implique un échantillonnage déterministe et une perte de variété.

Enfin, tous les algorithmes décrits plus haut présentent des performances similaires. Ils donnent d'excellents résultats pour des problèmes linéaires mais s'avèrent inefficaces pour des problèmes présentant de fortes interactions entre variables.

7.2 Modèles de dépendances deux à deux

L'UMDA permet d'approximer le comportement d'un algorithme génétique utilisant la recombinaison uniforme [Müh97] : il assure une reproduction et une ré-association adéquate pour des problèmes comportant des briques de base d'ordre 1. Il fonctionne donc très bien pour des problèmes dépourvus d'interactions significatives entre variables. Dans le cas contraire les BB d'ordre supérieur sont susceptibles d'être détruites d'une génération sur l'autre. Pour résoudre ce problème, des approches reposant sur des modèles plus riches ont été proposées. Ces algorithmes couvrent les interactions deux à deux entre ces variables.

L'algorithme MIMIC (*Mutual Information Maximizing input Cluctering*) [dBIJV97] recherche un ordonnancement des variables tel que les informations mutuelles calculées entre les variables voisines au sein de cet ordre soient maximales. Cette méthode permet de minimiser la divergence de Kullback-Liebler [KL51] entre la chaîne construite et la distribution jointe des variables au sein de la population. Nous parlons de chaîne dans la mesure où chaque gène virtuel ne dépend que de ces voisins au sein de l'ordonnancement choisi. L'inconvénient de cette approche est que la recherche d'un ordonnancement optimal des gènes virtuels requiert l'utilisation d'un algorithme glouton ne garantissant pas l'optimalité de la distribution apprise. Une généralisation de cette approche a été proposée par Baluja et Davies qui ont utilisé des arbres de dépendance afin de modéliser la distribution des solutions parents. Outre le fait que la méthode proposée

présente des garanties d'optimalité, les modèles appris sont surtout plus généraux que les modèles en forme de chaîne proposés par MIMIC. Pelikan et Muhlenbein ont proposé un modèle encre plus général : l'algorithme à distribution marginale bivariée ou BMDA (pour *Bivariate Marginal Distribution Algorithm*) [PM99]. Ce dernier permet de construire une forêt, c'est-à-dire un ensemble d'arbres mutuellement indépendants. Dans ces travaux, les dépendances entre variables sont déterminées grâce au test du χ^2 de Pearson [MM77].

Les modèles deux à deux permettent de couvrir des dépendances entre variables dans un problème d'optimisation et sont très faciles à apprendre. Les algorithmes qui viennent d'être présentés permettent de reproduire et d'échanger efficacement des BB d'ordre 2 et par conséquent ils donnent des résultats satisfaisants pour des problèmes linéaires ou quadratiques [dBIJV97, BD97, Müh97, PM99, BT00]. Toutefois cela demeure insuffisant pour envisager la résolution de problèmes comportant des interactions d'ordre supérieur entre leurs variables [PM99].

7.3 Modèles à dépendances multiples

L'utilisation de modèles plus généraux que ceux que nous avons présentés jusqu'à maintenant permet d'envisager des algorithmes capables de résoudre des problèmes décomposables de manière efficace. Cela implique cependant d'utiliser des algorithmes d'apprentissage complexes souvent lourds à mettre en œuvre et qui ne garantissent pas l'optimalité des modèles de distribution appris. Sur le plan calculatoire, cette option reste cependant intéressante dans la mesure où le temps accru consacré à l'apprentissage du modèle est (partiellement) compensé par une réduction significative du nombre d'évaluations de la fonction objectif. Les itérations sont plus longues mais moins nombreuses car chaque pas au sein de l'espace de recherche, bien que plus lent, est également plus « intelligent » : le modèle appris est censé guider l'algorithme évolutionnaire vers des régions de l'espace de recherche. Les algorithmes qui sont présentés par la suite exploitent une modélisation permettant de capturer les interactions multivariées.

7.3.1 L'algorithme génétique compact étendu (eCGA)

À chaque itération, l'algorithme génétique compact étendu (*extented compact genetic algorithm* [Har99]) range les variables au sein d'un ensemble de groupes considérés comme des BB. Les gènes virtuels appartenant à un groupe sont considérés comme dépendants alors que d'un groupe à l'autre ils sont considérés comme indépendants. Ces groupes sont ensuite traités comme le sont les variables uni-variées au sein de l'UMDA : pour chaque groupe identifié, la distribution jointe des variables qui le compose est estimée. Les distributions des différents groupes sont ensuite échantillonnées afin de générer les solutions candidates de la nouvelle population. La phase critique réside dans la construction des différents groupes à partir des solutions parents. Elle est réalisée grâce à une méthode d'apprentissage à base de score. Chaque partition des variables en sous-groupes mutuellement indépendants est évaluée d'après le score MDL. Ce dernier permet de mesurer l'adéquation du modèle reposant sur la partition proposée à la distribution des solutions dans la population $PopPar$. En outre, ce score pénalise les modèles trop complexes, c'est-à-dire les modèles proposant un petit nombre de groupes de variables de grande taille. La recherche du modèle minimisant le score MDL est réalisée par une procédure gloutonne : partant d'un ensemble de variables indépendantes, à chaque génération deux ensembles (pouvant chacun ne contenir qu'une seule variable) sont fusionnés afin de ne constituer qu'un seul et même groupe. Parmi toutes les fusions envisageables, on applique celle qui engendre le modèle de score minimum. Cette procédure itérative se poursuit jusqu'à ce qu'aucune minimisation du score ne soit

plus possible. D'après la théorie développée pour les UMDA, des problèmes séparables, c'est-à-dire décomposables en sous-problèmes non chevauchants d'ordre limité, peuvent être résolus dans un temps sub-quadratique. Cela implique cependant que l'eCGA parvienne à identifier un bon modèle. En outre, de nombreux problèmes présentent d'importants chevauchements entre leurs BB, il est alors difficile de modéliser correctement le problème par une simple partition des variables en jeu.

7.3.2 L'algorithme à distribution factorisée (FDA)

L'algorithme (*Factorized Distribution Algorithm* ou FDA [MM99]) utilise un modèle fixe — en l'occurrence une distribution factorisable — durant tout le processus d'optimisation. Le FDA n'apprend donc pas la structure du problème, la distribution et sa décomposition/factorisation lui étant préalablement fournies par un expert. La distribution résultante est donc un produit de distribution marginales et conditionnelles qui sont mises à jour à chaque génération d'après les solutions sélectionnées. Ce sont donc seulement les paramètres de la distribution qui sont estimés au cours de l'optimisation et non sa structure qui est prédéfinie. Mühlenbein et Mahnig [MM99] ont prouvé que si ce modèle est correct , FDA résout des problèmes décomposables efficacement, cependant l'information a priori relative à la structure du problème et nécessaire à son utilisation est rarement disponible. L'utilisation de cette approche est donc limitée à des problèmes déjà bien connus.

Pour remédier à ce problème, des algorithmes permettant d'apprendre en temps réel la structure du problème en mettant en œuvre des modèles graphiques capables de capturer des interactions complexes ont été développé.

7.3.3 L'algorithme d'optimisation Bayésien (BOA)

L'algorithme d'optimisation Bayésien (*Bayesian Optimization algorithm* ou BOA [PGCP99]) utilise une classe de distributions plus générale que l'eCGA. Il utilise un réseau Bayésien afin de modéliser la distribution des solutions prometteuses. Ce dernier est construit à partir des solutions sélectionnées en utilisant l'une des nombreuses méthodes d'apprentissage de réseaux Bayésiens disponibles. Le plus souvent il s'agit de méthodes d'apprentissage à base de score. N'importe quel critère de qualité peut être utilisé afin de discriminer les réseaux Bayésiens candidats : score BD (Bayesian Dirichlet), MDL (Minimum Description Length), BIC (Bayesian Information Criterion). Dans la plupart des publications cependant, le score BD est utilisé. De même, parmi les nombreuses méthodes d'exploration de l'espace des réseaux Bayésiens, les algorithmes de type montée de colline sont les plus fréquemment utilisés du fait, notamment, de leur rapidité et de leur simplicité. L'inconvénient majeur de cette recherche gloutonne vient du fait qu'il s'agit d'une méthode d'optimisation locale. Compte tenu de l'extrême complexité de l'apprentissage de structure dans les réseaux Bayésiens, on peut donc s'attendre à identifier un modèle sous-optimal qui malgré sa richesse de représentation ne capturera pas la structure du problème.
Afin de limiter la taille de l'espace de recherche et de simplifier l'apprentissage, il est possible de spécifier au préalable l'ordre de grandeur des interactions que l'on souhaite modéliser. En outre, l'usage de contraintes sur le degré entrant des noeuds dans un réseau Bayésien permet de palier le principal défaut du score BD évoqué plus haut en limitant de manière explicite la complexité des modèles appris.

La classe de modèles utilisée par le BOA est équivalente à celle mise en oeuvre dans le FDA (modèles factorisables en distributions conditionnelles et marginales). Toutefois, contrairement à

ce dernier, BOA ne requiert aucune connaissances *a priori* concernant la structure du problème puisqu'il la découvre par lui-même au cours du processus d'optimisation. Cependant, afin de guider la recherche du modèle optimal, il est possible de combiner des connaissances a priori relatives à la structure du problème avec l'information représentée par l'ensemble des solutions prometteuses. Le ratio entre ces deux types d'informations peut être contrôlé et l'apport de l'information *a priori* pondérée. Notons toutefois que si celle-ci est très utile pour parvenir à un apprentissage de qualité elle n'en est pas pour autant nécessaire. En effet comme nous l'avons évoqué précédemment, il est difficile de définir une distribution a priori sur la structure d'un réseau Bayésien sans s'autoriser des approximations fortes sur l'espace des structures (telles que la non prise en compte de la contrainte d'acyclicité). À structure fixée il est par contre plus aisé de définir un a priori sur le paramétrage du modèle.
Enfin, notons l'existence d'un second algorithme exploitant les réseaux Bayésiens pour modéliser la distribution des solutions prometteuses, appelé algorithme à estimation de réseaux Bayésiens (*Bayesian network estimation algorithm* ou EBNA) proposé par Etxeberria et collèges [ELP97].

7.4 Utilisation des EDA pour l'apprentissage de structure dans les réseaux Bayésiens

Avant toute chose, pour éviter toute confusion dans notre propos, précisons qu'à partir de maintenant nous parlerons de *modèle interne* pour désigner le modèle utilisé dans les EDA afin de modéliser la distribution des solutions parents. Nous parlerons de *modèle externe* pour désigner le modèle que nous souhaitons apprendre avec les algorithmes évolutionnaires, en l'occurrence un réseau Bayésien. De la même manière nous parlerons d'apprentissage interne ou externe selon le type de modèle auquel on se réfère.

Les EDA apparaissent comme une alternative intéressante aux algorithmes génétiques pour l'apprentissage de structure dans les réseaux Bayésiens. En effet, comme nous l'avons vu, les EDA se fondent sur une approche probabiliste pour explorer l'espace de recherche durant la phase de reproduction. Cela nous permet de bénéficier d'un ensemble de méthodes et de résultats théoriques propres à l'estimation de modèles probabilistes dont les AG sont pour leur part dépourvus. Ces derniers reposent il est vrai sur un très grand nombre de travaux proposant des solutions aux différents problèmes pouvant se poser dans le cadre d'un problème d'optimisation multi-modal. Cependant, lorsque l'on ne dispose d'aucune connaissance concernant la structure du problème d'optimisation, aucune solution satisfaisante n'existe aujourd'hui pour concevoir un AG générique qui puisse traiter des variables présentant des dépendances multiples. La grande force des EDA est la promesse de pouvoir apprendre en temps réel, durant l'optimisation, la structure du problème en identifiant les dépendantes entre variables. Bien sûr, de la promesse à la réalité le chemin est pavé d'embûches, et comme nous l'avons précisé plus tôt, l'estimation du modèle interne dans un EDA peut constituer en soi un vrai problème. Cela est particulièrement sensible avec le BOA. En effet, nous avons fondé notre travail sur l'idée que l'apprentissage de structure dans les réseaux Bayésiens était une tâche particulièrement difficile. Il peut donc paraître paradoxal de prétendre apprendre un réseau Bayésiens au sein même d'un EDA, avec une méthode aussi simple qu'un algorithme de montée de colline. À ce stade, deux réflexions peuvent être faites. Tout d'abord, à moins que les variables du problème d'optimisation ne présentent réellement un grand nombre d'interactions significatives, on peut se demander s'il est réellement nécessaire de recourir à des modèles internes très complexes à apprendre. Par ailleurs, même lorsque l'on utilise un modèle interne complexe tel qu'un réseau Bayésien, il est possible que l'on obtienne un résultat satisfaisant à l'issue de l'EDA même l'estimation du

modèle interne est imparfaite. Par exemple, on peut avancer que même si un réseau Bayésien ne capture pas toutes les dépendances entre les gènes virtuels, celles qu'il a identifiées peuvent malgré tout guider l'EDA et assurer de meilleurs résultats que ceux que nous aurions obtenus avec un modèle sans dépendances. La question du choix du modèle interne est donc ouverte. À notre connaissance, les travaux s'étant intéressé à l'utilisation des EDA pour l'apprentissage de structure ont surtout considéré des modèles sans dépendances. Peña et collègues [PnLLn04] ont utilisé des UMDA pour faire évoluer des structure de réseaux Bayésiens représentées dans un codage similaire à nos chromosomes relationnels. Dans leurs travaux, l'apprentissage de structure de réseaux Bayésiens n'était cependant pas une fin en soi. Il s'agissait d'utiliser les réseaux Bayésiens appris pour faire de la classification et l'exactitude de la structure apprise ne constituait donc pas un point central dans leur travaux. Blanco et collègues [BILn03] ont pour leur part considéré l'apprentissage de structure avec les algorithmes UMDA et PBIL, montrant que ces derniers parvenaient à maximiser les différents scores utilisés (vraisemblance marginalisée et maximum de vraisemblance pénalisée) de manière satisfaisante tout en offrant des temps de calcul plus intéressants qu'un algorithme génétique.

Dans ce qui suit, nous allons tester deux EDA diamétralement opposés afin de rendre compte de la pertinence des modèles internes complexes. Nous allons comparer l'UMDA qui est un EDA utilisant un modèle interne sans dépendance, avec un BOA, dont le modèle interne est le plus complexe à ce jour dans la littérature. Pour réaliser ces tests, nous nous plaçons dans des conditions rigoureusement identiques à celles que nous avons considérées pour tester nos différentes stratégies évolutionnaires dans le chapitre 6.

7.4.1 Utilisation des algorithmes UMDA et BOA

Tout d'abord, les deux EDA testés sont appliqués à des chromosomes relationnels. Nous rappelons que les gènes virtuels composant les chromosomes relationnels sont des variables ternaires. Chacune d'entre elles représente la relation existant entre deux sommets X et Y du DAG candidat (modèle externe) : X est indépendant de Y, $X \rightarrow Y$ ou $X \leftarrow Y$.

Dans les grandes lignes, ces deux EDA fonctionnent selon le même synopsis que l'algorithme évolutionnaire que nous avons proposé :

Initialisation de la population Un ensemble $Pop(t)$ comprenant M DAG est généré aléatoirement. Les DAG sont évalués grâce au score BIC.

Répéter jusqu'à ce qu'un critère d'arrêt soit satisfait :

1. Tirer les $M/2$ DAG les plus performants au sein de la population $Pop(t)$ pour constituer la population $PopPar(t)$,
2. Apprendre un modèle interne $h \in H$ représentant la distribution des solutions candidates dans $PopPar(t)$ où :
   - pour UMDA : h est un ensemble de distributions marginales, chacune caractérisant la distribution des valeurs prises par un gène virtuel dans $PopPar(t)$;
   - pour BOA : h est un réseau Bayésien dont la structure est un DAG ayant pour sommets les gènes virtuels des individus de la population.
3. Échantillonner h afin de générer une population de descendants appelée $PopEnf(t)$ comprenant M individus.
4. Les individus de $PopEnf(t)$ sont réparés pour respecter la contrainte d'acyclicité ainsi que la contrainte sur le degré entrant maximum des sommets (fixé à 10).

5. Le score BIC des DAG de $PopEnf(t)$ est calculé.
6. Les descendants constituent la nouvelle population à la génération suivante : $Pop(t+1) \leftarrow PopEnf(t)$.

Cette fois, l'algorithme ne fonctionne pas selon un mode stationnaire et l'ensemble des individus de la population est remplacé d'une génération sur l'autre. De même, il n'y a pas de mutation, l'étape d'échantillonnage générant déjà une variabilité suffisante. La phase de reproduction diffère en fonction de l'algorithme utilisé. Dans le cas de l'UMDA, nous nous contentons d'apprendre les fréquences des trois configurations de chacun des gènes virtuels dans $PopPar(t)$. Dans le cas du BOA, il nous faut apprendre à la fois la structure du modèle interne et ses paramètres. On comprend que le problème d'apprentissage de structure est particulièmerent périlleux. En effet, le nombre de sommets du DAG du modèle interne est égal au nombre de gène virtuels. Or, ces derniers correspondent aux relations entre les sommets du DAG du modèle externe. La taille des modèles internes appris par le BOA est donc très importante, au point que l'apprentissage de structure risque d'être impraticable dans un temps raisonnable.

Exemple 7.1
Si nous souhaitons apprendre un réseau Bayésien de 10 sommets avec un BOA, nous devons construire une population de chromosomes codant des DAG candidats avec 45 ($\frac{10^2-10}{2}$) gènes virtuels. À chaque génération, le BOA essaie de construire un réseau Bayésien comportant 45 sommets afin de capturer les dépendances entre ces gènes virtuels. Pour apprendre un réseau Bayésien comportant 10 variables, nous devons donc en apprendre un qui en comporte 45 !

Quelques restrictions ont donc été entreprises pour pouvoir apprendre ces DAG dans de bonnes conditions. En première approche, nous avons envisagé de restreindre la recherche de structure à l'espace des arbres orientés. Sous ces conditions, il est possible d'appliquer l'approche de Chow et Liu [CL68] et d'utiliser la technique de l'arbre de recouvrement maximum qui est à la fois rapide et efficace. Cependant on se ramène alors à un modèle de dépendance deux à deux. Afin d'enrichir ce modèle, nous avons donc décidé d'appliquer à l'arbre produit par cette méthode d'apprentissage un algorithme de montée de colline. Cependant, pour éviter que les temps de calcul ne soient rédhibitoires nous avons imposé une contrainte très forte sur cette seconde étape : nous avons fixé un nombre d'itérations maximum très faible à l'algorithme de montée de colline. Nous avons décidé qu'il ne réaliserait que 100 itérations à chaque fois. Cela est peu mais il est ainsi possible d'affiner le modèle obtenu par l'ajout ou la suppression de quelques arcs. Surtout, nous garantissons ainsi un temps de calcul raisonnable pour mener nos tests.

La dernière remarque concerne le processus de réparation. Ce dernier s'applique au BOA de la même manière qu'aux AG. Après avoir échantillonné un ensemble d'arcs à partir de la distribution de probabilité modélisée par le réseau Bayésien, le nouvel individu est construit de manière itérative : chaque arc est testé avant de venir s'y insérer. Ce test permet de s'assurer que l'arc introduit dans le DAG en construction respecte la contrainte d'acyclicité ainsi que la contrainte sur le degré entrant maximum qui est toujours fixé à 10. Si un arc viole l'une de ces contraintes, il est inversé et on teste son addition dans le sens opposé. La différence entre le BOA et l'UMDA se joue au niveau de l'ordre dans lequel ces arcs sont testés. Dans le cas du BOA, les arcs sont pris dans un ordre aléatoire, comme c'était le cas pour les algorithmes génétiques. Pour l'UMDA, nous disposons directement de la probabilité marginale de chacun de ces arcs au sein du modèle interne. Nous les ordonnons donc en fonction de cette probabilité, les plus probables étant introduits en premier dans le DAG enfant. Grâce à cela, les arcs les plus probables ont moins de chance d'être inversés ou supprimés, la probabilité de créer un cycle ou de pointer sur un sommet saturé augmentant avec le nombre d'arcs déjà présents dans le graphe.

Notons que l'approche consistant à ne conserver que les graphes échantillonnés respectant ces contraintes présente deux inconvénients qui la rendent moins intéressante que notre méthode de réparation. Le premier inconvénient est un coût calcul plus important car une forte proportion des graphes générés présente des cycles. Par conséquent, il est généralement nécessaire de réaliser beaucoup de rejets avant d'obtenir une solution admissible. Ensuite, une telle approche revient également à biaiser la phase d'échantillonnage. En effet, si les arcs $a \rightarrow b$, $b \rightarrow c$, et $c \rightarrow a$ ont tous les trois une probabilité très élevée, on ne respecte pas non plus la loi du modèle interne car les graphes présentant ces trois arcs ont une probabilité nulle d'être retenus.

7.4.2 Comparaison des deux approches

Les tests qui suivent ont été réalisés selon un protocole similaire à celui que nous avons appliqué aux algorithmes génétiques étudiés au chapitre précédent. Chaque EDA est utilisé pour apprendre la structure d'un réseau Bayésien pré-déterminé (en l'occurrence le réseau Insuline). Pour cela, on lui soumet une base d'apprentissage de taille fixe, comptant 300 mesures de l'état des 35 variables constitutives du modèle. Chacune de ces mesures est obtenue par échantillonnage du réseau objectif. Le score utilisé pour évaluer la qualité des modèles candidats et constituant la fonction objectif à optimiser est de nouveau le score d'information Bayésien (BIC).
Le DAG codant la structure du meilleur réseau Bayésien appris à chaque itération par les EDA est converti en CPDAG. Il en va de même pour le DAG encodant la structure du réseau Bayésien dont nous avons extrait les données. Deux mesures de qualité sont extraites de la comparaison des deux CPDAG résultants :

- la *sensibilité* rend compte de la proportion des connections entre variables appartenant au graphe objectif que nous sommes parvenu à identifier ;
- la *ppv* rend compte de la proportion de vrais positifs, c'est-à-dire d'arcs appartenant effectivement au modèle objectif, parmi tous les arcs identifiés lors de l'apprentissage.

Chaque test étant réalisé 10 fois sur 10 jeux de données distincts et indépendants, les courbes que nous présentons représentent donc la valeur moyenne ainsi que l'écart type de l'une de ces deux mesures de qualité durant le déroulement d'un EDA donné.

7.4.2.1 Comparaison de l'UMDA et du BOA pour une population de 200 DAG

Dans un premier temps, nous avons testé ces deux EDA avec une taille de population similaire à celle utilisé par les algorithmes génétiques dans le chapitre précédent.

Rappelons que dans le cadre des EDA, la taille de la population correspond à la taille de l'échantillon de solutions candidates utilisé pour apprendre le modèle interne.
Les figures 7.3 et 7.1 représentent les traces moyennes ainsi que les écarts type de la sensibilité pour un UMDA et un BOA utilisant une population de 200 individus. Comme on peut le constater, bien que les deux donnent des résultats médiocres on remarque que l'UMDA atteint une sensibilité de $0,5$ alors que la BOA ne dépasse pas les $0,35$. Les figures 7.4 et 7.2 représentent les traces moyennes ainsi que les écarts type de la ppv pour l'UMDA et l'BOA respectivement. Les résultats obtenus en termes de ppv sont par contre surprenants. Alors que le BOA demeure sous la barre des $0,5$, l'UMDA affiche une ppv proche de $0,8$. Dans l'absolu, compte tenu de la faible sensibilité des DAG produits par cet algorithme, on est en droit de supposer que ces derniers présentent peu de faux positifs parce qu'ils sont très parcimonieux. Il est cependant intéressant de remarquer qu'outre de meilleurs résultats, l'UMDA semble plus robuste que le BOA, l'écart type des résultats obtenus sur les 10 bases d'apprentissage étant plus faible pour l'UMDA.

7.4.2.2 Comparaison de l'UMDA et du BOA pour une population de 2000 DAG

Afin d'expliquer les faibles performances du BOA, nous avons entrepris d'utiliser des populations de taille nettement supérieure. Les mauvais résultats du BOA peuvent s'expliquer par le fait que l'apprentissage du modèle interne nécessite une base d'apprentissage — et donc un nombre de solutions parents — plus grande. Le nombre de sommets du DAG construit par le BOA est en effet considérable. Le problème d'optimisation externe comporte 35 variables, une par sommet du DAG que nous voulons apprendre. Chacun des DAG candidats est codé par un chromosome comportant 595 gènes virtuels (ce chiffre correspondant au nombre de paires de sommets au sein du DAG initial). Par conséquent le modèle interne construit par le BOA est un réseau Bayésien de 595 variables ternaires. Plutôt que de chercher à utiliser une méthode d'apprentissage élaborée coûteuse en termes de calcul, sans garantie de résultat, il nous a semblé préférable d'augmenter la taille de la population afin d'aider le BOA à mieux identifier les dépendances entre les gènes virtuels.

Les figures 7.7 et 7.5 confirment nos soupçons. Avec une base d'apprentissage plus large, le BOA parvient à se rapprocher de l'UMDA qui naturellement, donne de meilleurs résultats que dans le cas précédent. En fin d'évolution, les deux algorithmes ont une sensibilité proche de $0,65$. En outre, on peut noter que la variabilité des résultats sur les 10 jeux de données est relativement faible. Ces observations sont confirmées dans les figures 7.8 et 7.6 qui montrent la ppv des DAG appris par ces deux algorithmes. Dans les deux cas, elle est proche de $0,9$. Cela implique un taux de faux positifs extrêmement faible et est de très loin le meilleur résultat que nous puissions espérer concernant cet indice de qualité. Bien sûr, la sensibilité est en retrait. Malgré tout, les résultats obtenus par ces deux algorithmes restent du même ordre que ceux obtenus avec l'algorithme génétique testé au chapitre précédent. Ce dernier atteignait une sensibilité et une ppv légèrement inférieure et supérieure à $0,7$ respectivement. Les EDA que nous venons de tester présentent donc une ppv supérieure et une sensibilité inférieure. Il faut cependant noter que les EDA bénéficient ici d'une taille de population très supérieure à l'AG. On peut supposer qu'avec une population plus importante ce dernier pourrait également donner de meilleurs résultats. Cette éventualité est cependant peu réaliste pour des questions de temps de calcul. En moyenne, notre AG mettait trois heures pour achever ses calculs. Avec une population de 2000 individus, le BOA met près de 9 heures et l'UMDA de l'ordre de 4 heures. Avec une population de 200 individus le BOA et l'UMDA mettaient respectivement deux heures et une heure en moyenne pour converger. Comme on peut le constater les temps de calcul auxquels nous faisons référence sont importants et il peut être gênant d'augmenter la taille de la population pour un algorithme tel que le BOA.

On notera qu'en comparaison du temps de calcul, le nombre de générations varie assez peu d'une condition à l'autre. Pour une population de 200 individus, l'UMDA et le BOA s'étendent en moyenne sur 56 et 46 générations respectivement. Pour une population de 2000 individus, l'UMDA et le BOA convergent en moyenne après 60 et 66 générations respectivement. La phase d'apprentissage du modèle interne est donc clairement responsable des importantes variations en terme de temps de calcul. On constate également que l'UMDA est moins sensible à l'augmentation de la taille de la population que le BOA. En effet, avec 200 individus, le BOA nécessite moins de générations pour converger que l'UMDA. Ce rapport de force est inversé lorsque la taille de la population passe à 2000.

Au final, il apparaît que seul l'UMDA est suffisamment rapide pour supporter une augmentation de la taille de la population. Au vu des résultats obtenus durant ces tests, il semble qu'il s'agisse d'une méthode prometteuse. On peut supposer qu'une version plus élaborée de ce dernier, utilisant une technique similaire au niching, aurait surpassé l'algorithme génétique que nous avons présenté.

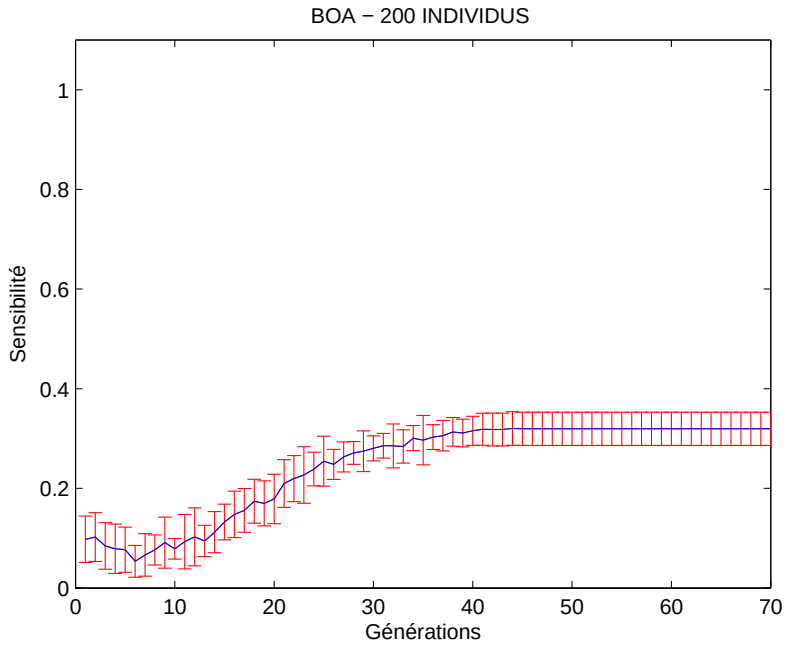


Fig. 7.1 – Evolution de la **sensibilité** au fil des générations d'un **BOA** pour une population de **200 individus**.

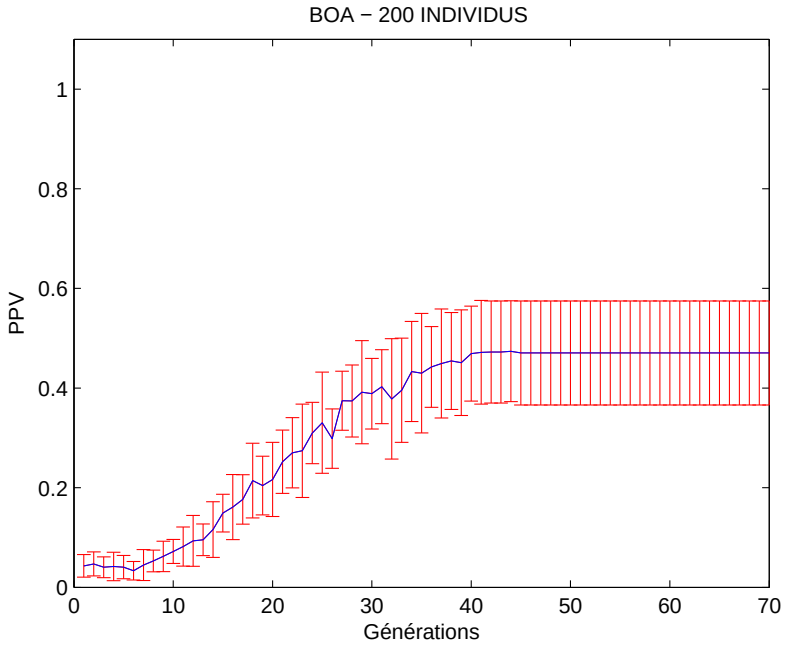


FIG. 7.2 – Evolution de la **PPV** au fil des générations d'un **BOA** pour une population de **200 individus**.

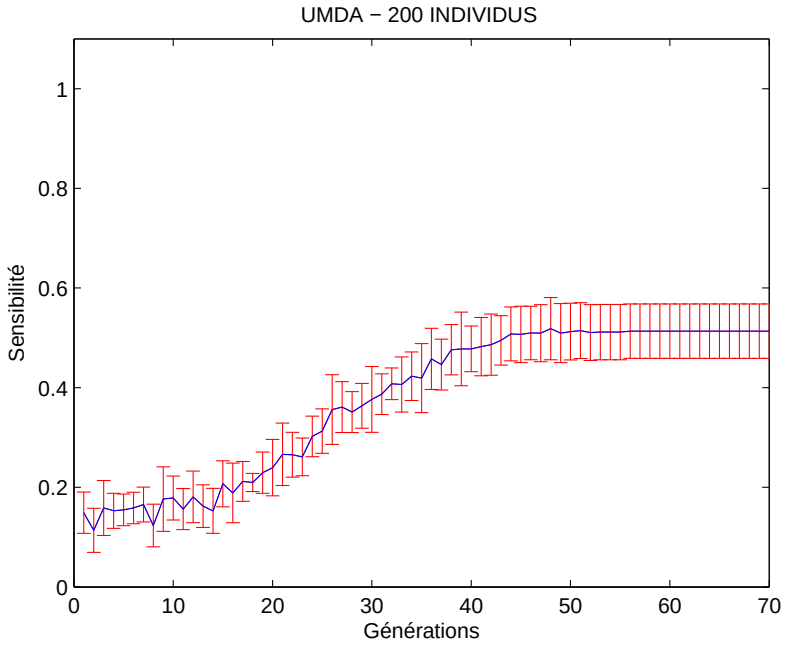


FIG. 7.3 – Evolution de la **sensibilité** au fil des générations d'un **UMDA** pour une population de **200 individus**.

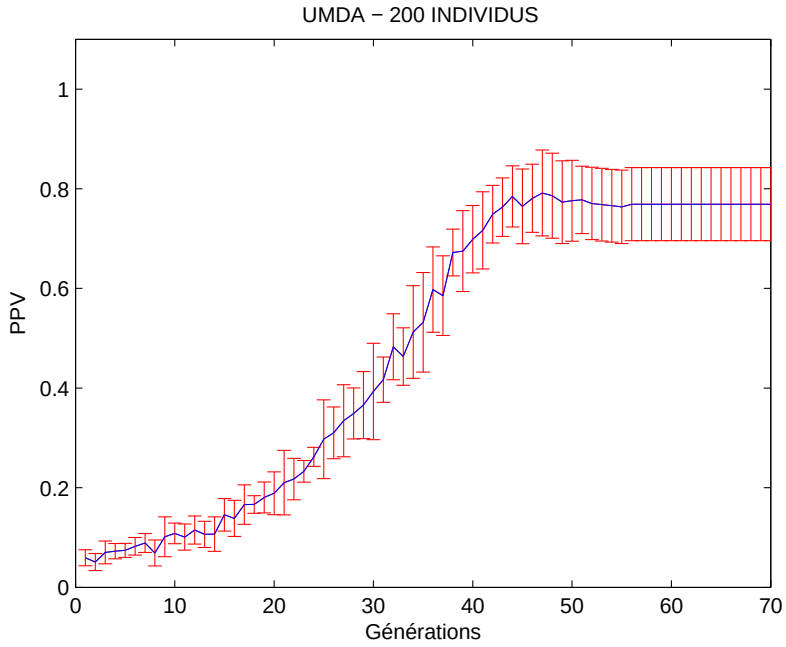


FIG. 7.4 – Evolution de la **PPV** au fil des générations d'un **UMDA** pour une population de **200 individus**.

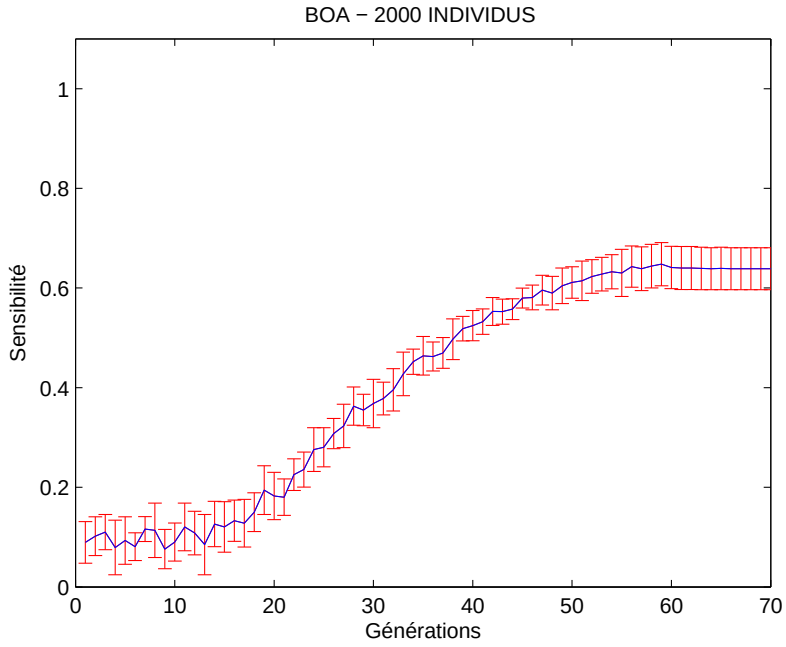


FIG. 7.5 – Evolution de la **sensibilité** au fil des générations d'un **BOA** pour une population de **2000 individus**.

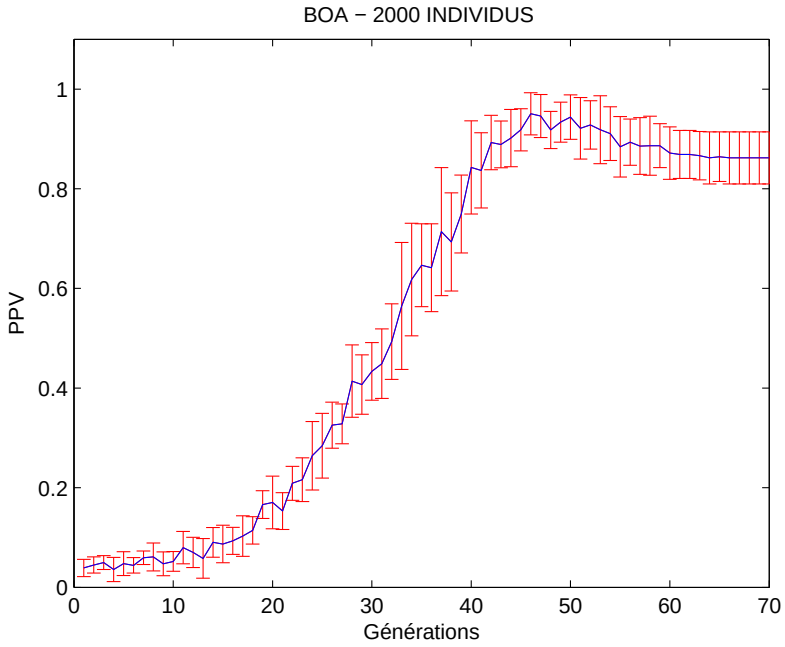


FIG. 7.6 – Evolution de la **PPV** au fil des générations d'un **BOA** pour une population de **2000 individus**.

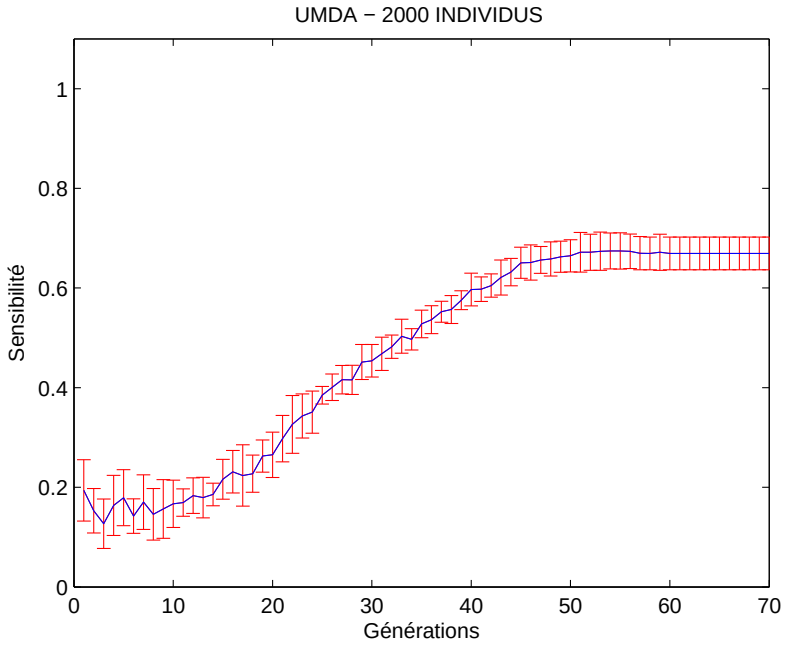


FIG. 7.7 – Evolution de la **sensibilité** au fil des générations d'un **UMDA** pour une population de **2000 individus**.

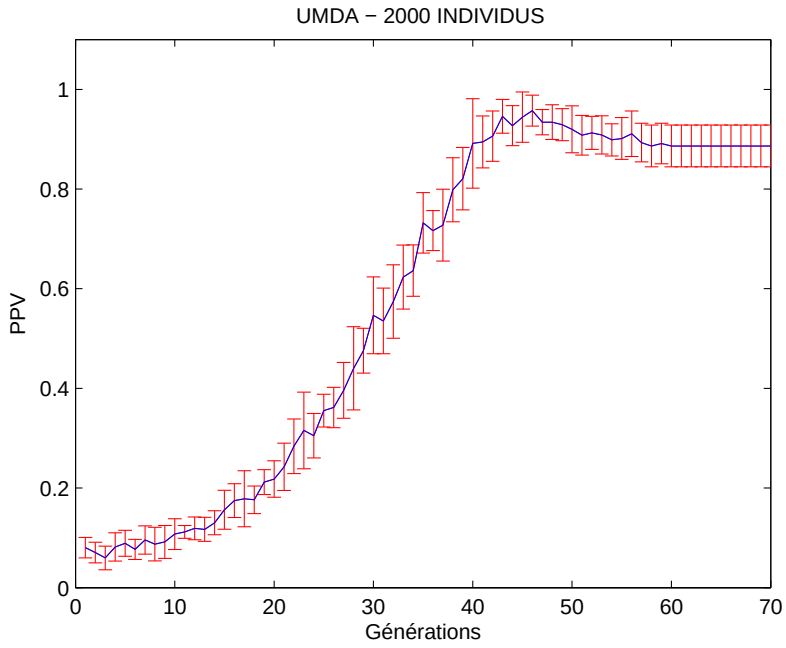


FIG. 7.8 – Evolution de la **PPV** au fil des générations d'un **UMDA** pour une population de **2000 individus**.

Conclusions et perspectives

Conclusions

Les travaux présentés dans cette thèse ont permis de démontrer l'intérêt des approches évolutionnaires pour l'apprentissage des réseaux de régulation transcriptionnelle.

Dans la première partie de ce mémoire, nous avons montré que la notion de réseau de régulation génétique constituait un cadre intéressant pour l'étude des grandes fonctions cellulaires. Plus particulièrement, nous avons porté notre attention sur les réseaux de régulation transcriptionnelle. En effet, de nombreux phénomènes cellulaires sont contrôlés par l'action concertée des gènes qui modulent la quantité de transcrits à l'équilibre pour une cellule. De plus, la régulation transcriptionnelle présente l'avantage de pouvoir être étudiée à l'échelle du génome grâce à la technique des puces à ADN. Ces dernières génèrent cependant des données complexes présentant une grande variabilité. Nous avons donc présenté les nombreuses méthodes d'analyse de données qui permettent d'extraire, à partir des mesures de profils d'expression, des listes de gènes qui sont dérégulés ou qui présentent des profils similaires. Cependant, pour extraire des informations concernant l'activité coordonnées des gènes, il apparaît nécessaire de se tourner vers les méthodes de représentation ou d'apprentissage de modèles. Ceux-ci permettent de traduire ou d'extraire des connaissances concernant les mécanismes de régulation qui sont à l'œuvre dans la cellule.

Dans la seconde partie de ce mémoire, nous avons comparé divers formalismes mathématiques quant à leur intérêt pour décrire des influences régulatrices entre les gènes observés ou quant à leur capacité à rétro-concevoir (reverse engineering) les influences régulatrices entre les gènes observés à partir des données d'expression. Nous nous sommes tournés vers les modèles graphiques et plus particulièrement vers les réseaux bayésiens qui offrent un cadre riche pour l'apprentissage des réseaux de régulation. En effet, il s'agit de modèles probabilistes adaptés à la modélisation de données bruitées, qui prennent en compte la nature stochastique des phénomènes biologiques étudiés. Ils ont fait l'objet de nombreuses études et bénéficient ainsi d'un nombre conséquent d'outils et de résultats théoriques. Cela permet d'envisager l'apprentissage de structure de ces réseaux bayésiens dans le cadre très difficile de la génomique fonctionnelle : le nombre de mesures disponibles sera toujours très faible comparé à la taille des échantillons habituellement utilisés en apprentissage artificiel. Parmi les différentes approches permettant de faire de l'apprentissage de structure dans les réseaux bayésiens, nous avons concentré notre attention sur les méthodes d'exploration de l'espace des structures. Ces dernières reposent sur la définition d'un score permettant d'évaluer la capacité d'un modèle à rendre compte des données ainsi que sur

une méthode de recherche permettant d'identifier le modèle maximisant ce score. Compte-tenu de la complexité de cette tâche, nous avons mis en avant la nécessité de recourir à des heuristiques stochastiques afin d'assurer un parcours efficace de l'espace des solutions candidates.

Dans ce but, nous avons proposé d'utiliser des algorithmes évolutionnaires. Ces derniers simulent l'évolution d'une population de solutions candidates à travers un processus itératif de recombinaison et de sélection de ces solutions. Nous avons comparé différentes méthodes de recombinaison afin d'étudier leurs performances en terme d'exploration de l'espace des DAG. Nous avons également testé l'apport d'une méthode de spéciation appelée *Deterministic Crowding* que nous avons confrontée à l'opérateur de mutation. Ces deux techniques visent à maintenir la diversité des solutions candidates au sein de la population. Elles tendent à améliorer le processus d'évolution en empêchant l'algorithme de tomber prématurément dans un minimum local. Afin de tester ces différentes approches évolutionnaires, nous avons choisi d'utiliser des données synthétiques échantillonnées à partir d'un réseaux bayésiens de 35 variables : le réseaux Insulin. Il s'agit d'un modèle artificiel bio-réaliste représentant l'homéostasie du glucose. Le modèle à l'origine des données étant connu, nous avons été en mesure d'évaluer les performances de chaque méthode d'apprentissage testée pour retrouver les arêtes du réseau original. Afin de nous placer dans des conditions réalistes, nous nous sommes limités à des bases d'apprentissage de petite taille que nous avons évaluées systématiquement. Alors que la plupart des travaux portant sur l'apprentissage de structure considèrent des jeux de données comportant plusieurs milliers d'observation, nous avons choisi de ne considérer que des échantillons comportant de 50 à 300 cas pour des réseaux de 35 sommets. Dans ces conditions, les tests que nous avons menés nous ont permis de sélectionner une approche évolutionnaire particulièrement prometteuse. Nous avons pu constater qu'une méthode de croisement reposant sur la recombinaison uniforme de chromosomes codant les interactions élémentaires entre les variables du problème, assurait une exploration efficace de l'espace de recherche. Nous avons également confirmé l'utilité de la mutation afin d'introduire de la diversité dans la population pour échapper à certains optima locaux. Surtout, nous avons démontré la nécessité de recourir à des méthodes de spéciation pour mener à bien ce problème d'optimisation complexe. Nous montrons que le Deterministic Crowding permet d'améliorer significativement les résultats. En outre, nous avons mis en évidence grâce à des méthodes de visualisation comme le Sammon mapping ou l'ACP kernelisée, que cette technique ne se contente pas de ralentir la convergence de l'algorithme évolutionnaire : elle permet également -dans un certain nombre de cas- d'entretenir un population hétérogène répartie dans des régions distinctes de l'espace de recherche. Nous avons ensuite comparé ces approches en les confrontant à un panel d'algorithmes d'apprentissage couramment utilisés dans la littérature : l'algorithme de monté de colline (greedy search), K2, MCMC, BN-PC et MMHC. Ces derniers couvrent un large spectre de méthodes d'apprentissage de structure et implémentent diverses stratégies comme l'exploration de l'espace des DAG (K2 et l'algorithme de monté de colline), mais aussi des approches hybrides fondées sur de l'apprentissage par contrainte (BN-PC et MMHC) ou encore des méthodes d'échantillonnage de l'espace des DAG (MCMC). Au final, notre algorithme évolutionnaire s'est montré capable de surpasser tous ces algorithmes, K2 mis à part. Cette exception n'est pas choquante dans la mesure où K2 a à sa disposition l'ordre topologique de sommets du graphe objectif. Cette connaissance *a priori* constitue un avantage déterminant dans une situation où les données sont peu nombreuses.

Pour achever ces travaux nous avons rapporté une étude exploratoire concernant l'utilisation des algorithmes à estimation de distribution (EDA) afin de remplacer les algorithmes génétiques pour l'apprentissage de structure de réseaux bayésiens. Les EDA constituent une famille d'approches évolutionnaires fondée sur des approches probabilistes : dans un EDA, les opérateurs de variations sont remplacés par une phase d'apprentissage de la distribution des solutions dans

la population suivie d'une phase d'échantillonnage de cette distribution permettant de générer de nouvelles solutions candidates. Les EDA se distinguent avant tout par le type de modèle probabiliste utilisé pour représenter la distribution des chromosomes au sein de la population. Nous avons choisi de comparer deux algorithmes radicalement opposés : UMDA et BOA. UMDA ignore les dépendances entre les gènes virtuels des chromosomes. Il représente la distribution de ces derniers comme le produit des distributions marginales des gènes virtuels. BOA modélise des dépendances d'ordre élevé entre les gènes virtuels au moyen d'un nouveau réseau bayésien (remarquons que dans ce réseau, les sommets sont les arêtes des réseaux de régulation génétique). Dans l'approche UMDA, le modèle est facile à apprendre puisqu'il s'agit de calculer pour chaque gène virtuel les fréquences d'apparition de chacun de ses états. Pour BOA, la tâche est plus complexe car comme nous l'avons vu, apprendre la structure d'un réseau bayésien est une tâche difficile. Au final, afin de simplifier le problème, nous avons choisi de restreindre l'espace des structures des réseaux bayésiens, à l'espace des arbres orientés. Il est ainsi possible d'utiliser la méthode de l'arbre de recouvrement maximum de Chow et Liu - très rapide - pour apprendre la structure du modèle. Ces deux algorithmes ont été confrontés au même type de données que les premiers algorithmes évolutionnaires testés. Nous avons ainsi pu constater que l'UMDA donnait des résultats satisfaisants pour 300 données alors que le BOA produit des solutions médiocres. Nous supposons que la faible taille de la population est responsable de ces mauvais résultats. En effet, l'utilisation de populations de taille plus importante (2000 solutions candidates contre 200 précédemment) permet d'améliorer les performances du BOA mais au prix d'un temps de calcul plus conséquent. A l'issue de ces tests l'UMDA apparaît donc comme une alternative prometteuse aux algorithmes génétiques.

Perspectives

A l'issue de ce travail, plusieurs pistes s'offrent à nous.

La plus évidente concerne le contexte de mise en œuvre des algorithmes génétiques. De la même manière qu'une initialisation judicieuse au moyen de l'algorithme de Chow et Liu permet d'améliorer significativement l'algorithme de monté de colline, il est possible d'initialiser un algorithme évolutionnaire avec un algorithme MCMC, en utilisant un échantillon de solutions prometteuses générées par ce dernier. Il est également possible d'envisager l'utilisation de méthodes de spéciation plus élaborées. De nombreux auteurs ont suggéré que les distances phénotypiques entre individus sont plus intéressantes que les distances génotypiques. Nous pourrions utiliser le noyau entre graphes présenté dans le cadre de l'ACP kernelisée pour calculer les similarités entre DAG candidats au sein même d'une méthode de spéciation. Cette métrique pourrait être utilisée par un algorithme de classification non supervisée tel qu'un algorithme de classification spectrale afin de déterminer les niches au sein de la population. En ce qui concerne ces deux premières idées de travail, la principale limitation est d'ordre calculatoire. Les algorithmes évolutionnaires impliquent des temps de calcul considérables. Si l'on rajoute à ces derniers, les temps d'exécution d'un algorithme MCMC ou d'un algorithme de classification appliqué durant tout ou partie du processus d'évolution , les temps de calculs ne seront pas réalistes en l'état. Pour entreprendre de tels travaux, il est nécessaire de passer par une phase de développement afin d'implanter un algorithme évolutionnaire optimisé dans un environnement plus rapide que Matlab.

Comme nous l'avons déjà laissé entendre, une direction de travail intéressante concerne l'amélioration des EDA. En particulier, nous pensons envisageable d'utiliser les outils de l'apprentissage pour résoudre efficacement des problèmes posés par les approches évolutionnaires. Un exemple emblématique de ces questions est l'utilisation de modèles de mélange pour modéliser différentes sous-populations de solutions occupant des niches distinctes. Un second exemple est

la possibilité de conserver un codage explicite des graphes candidats tout en utilisant des lois paramétriques ou semi-paramétriques de graphes aléatoires s'inspirant des modèles d'Erdös-Rényi [ER59] ou les mélanges de ces lois [PDR08] jusqu'ici utilisés pour l'analyse d'un seul graphe. Deux problèmes sont alors à résoudre : l'apprentissage de ces lois qui nécessite une taille d'échantillon (nombre de graphes) importante et l'échantillonnage de graphe aléatoire qui est un domaine assez nouveau. Si de plus, des résultats de convergence peuvent être obtenus, alors cette méthode pourrait constituer une voie très prometteuse non seulement pour l'apprentissage de réseaux de régulation mais aussi pour l'apprentissage de tout type de réseaux, dans un cadre probabiliste et statistique.

Une troisième direction de travail concerne l'application de notre algorithme d'apprentissage de structure à d'autres types de modèles : en effet, notre algorithme prend en entrée l'ensemble des données d'apprentissage et fait appel à un algorithme d'apprentissage de paramètres (par exemple ceux des tables de probabilités conditionnelles du réseau bayésien). Rien ne s'oppose donc à l'application de cette méthode à l'apprentissage de structure dans d'autres types de modèles : systèmes d'équations différentielles non linéaires ou réseaux bayésiens dynamiques non linéaires. Pour tous ces modèles, il est intéressant de distinguer l'apprentissage des paramètres pour une structure de graphe donnée, et l'apprentissage de la structure elle-même. Des algorithmes d'estimation de paramètres dans des modèles dynamiques non linéaires ont été développés dans l'équipe [QBdB07], à IBISC, et il serait possible de coupler les deux types d'algorithmes. Notons que, dans le cadre dynamique, la difficulté liée à la contrainte du DAG disparaît. Cependant, des limitations d'ordre pratique compliquent singulièrement de tels travaux : compte-tenu des problèmes d'identifiabilité des systèmes dynamiques, en l'absence de contraintes ou de connaissances *a priori*, il n'est possible aujourd'hui d'estimer les paramètres que pour des systèmes dynamiques non linéaires de petite taille. Pour que l'apprentissage de structure soit intéressant, il faudra donc que ces méthodes soient améliorées de manière à pouvoir envisager l'apprentissage de réseaux de quelques dizaines de variables. Ceci fait actuellement l'objet d'études dans la communauté. Enfin, les méthodes d'estimation de paramètres des modèles non-linéaires étant beaucoup plus lourdes à mettre en œuvre que l'estimation des paramètres d'un modèle linéaire ou d'une table de probabilité conditionnelle, elles aboutissent à des temps de calcul importants, même pour des modèles de taille réduite. La résolution ce type de problème devrait pourtant nous permettre d'envisager l'ultime objectif de l'apprentissage de réseaux de régulations à partir de cinétiques d'expression.

Glossaire

Différentiation Mécanisme permettant à un type de cellules — généralement appelées cellules souches — de se transformer en plusieurs types cellulaires spécialisés, c'est-à-dire capables de remplir des fonctions distinctes et spécifiques, propres à un tissu (par exemple la peau, le sang).

Division cellulaire Processus permettant aux cellules de se multiplier, chacune se divisant en deux cellules identiques entre elles et à la cellule « mère ». Ce processus appelé *mitose* comporte diverses phases cruciales pour la cellule parmi lesquelles la réplication de l'ADN. Cette étape consiste pour la cellule mère à dupliquer son matériel génétique afin que de transmettre à chacune des cellules filles un exemplaire de son génome.

ARN ribosomique Molécule entrant dans la composition des ribosomes, de vastes édifices moléculaires faisant office de chaîne de montage pour les protéines.

Micro-ARN des ARN simple brin longs d'environ 21 à 24 nucléotides présents dans la plupart des organismes pluricellulaires. Les miRNA sont des répresseurs post-transcriptionnels : en s'appariant à des ARN messagers, ils guident leur dégradation ou la répression de leur traduction en protéine.

ARN interférant Petit ARN double brin de 21 nucléotides pouvant se lier spécifiquement à une séquence d'ARN messagers grâce à un complexe protéique nommé *RISC* (pour *RNA induce silencing complex*). Ils peuvent alors empêcher l'expression de gènes en clivant cet ARN grâce à la protéine *Ago*, une endonucléase faisant partie du complexe RISC.

Puissance d'un test statistique Probabilité de rejeter une hypothèse nulle qui est fausse. Par exemple, dans le cas où l'hypothèse nulle avance qu'un gène n'est pas modulé entre deux conditions expérimentales, cela revient à déclarer un gène comme différentiellement exprimé quand il l'est réellement.

Erreur de type I Faux positifs, ou rejeter l'hypothèse nulle alors qu'elle est exacte. Par exemple, déclarer un gène comme différentiellement exprimé alors qu'il ne l'est pas.

Erreur de type II Faux négatifs, ou ne pas rejeter une hypothèse nulle qui est fausse. Par exemple ne pas déclarer un gène comme différentiellement exprimé alors qu'il l'est.

Sur-apprentissage Le sur-apprentissage arrive lorsqu'un modèle excessivement complexe, comprenant trop de paramètres, est construit à partir d'une petite base d'apprentissage. Le modèle classe remarquablement bien ces données, mais donnera de mauvais résultats avec de nouveaux jeux de données, appelés *base de test*, distincts de celui utilisé pour l'apprentissage.

Métabolome Le métabolome est constitué de l'ensemble des petites molécules, les métabolites, qui peuvent être trouvées dans un échantillon biologique, telles que les acides aminés, les sucres ou les hormones.

Protéome Le protéome est l'ensemble des protéines produites par un génome (chez l'homme environ 60 000) dans des conditions données, à un moment donné.

Transcriptome Le transcriptome est l'ensemble des ARN messagers (molécules servant de matrice pour la synthèse des protéines) issu de l'expression d'une partie du génome d'un tissu cellulaire ou d'un type de cellule.

Bibliographie

[ABC+06] Jamil Ahmad, Gilles Bernot, Jean-Paul Comet, Didier Lime, and Olivier Roux. Hybrid modelling and dynamical analysis of gene regulatory networks with delays. *ComPlexUs*, 3(4) :231–251, 2006.

[ACPS06] David B. Allison, Xiangqin Cui, Grier P. Page, and Mahyar Sabripour. Microarray data analysis : from disarray to consolidation and consensus. *Nature Reviews Genetics*, 7(1) :55–65, 2006.

[AdBF07] Cédric Auliac, Florence d'Alché Buc, and Vincent Frouin. Learning transcriptional regulatory networks with evolutionary algorithms enhanced with niching. In *Applications of Fuzzy Sets Theory, 7th International Workshop on Fuzzy Logic and Applications (WILF 2007)*, volume 4578 of *Lecture Notes in Computer Science*, pages 612–619. Springer, 2007.

[AFdB07] Cédric Auliac, Vincent Frouin, and Florence d'Alché Buc. Approches évolutionnaires pour l'apprentissage de réseaux de régulation transcriptionnelle. In *Conférence francophone sur l'Apprentissage automatique (CAp)*, 2007.

[AFGdB08] Cédric Auliac, Vincent Frouin, Xavier Gidrol, and Florence d'Alché Buc. Evolutionary approaches for the reverse-engineering of gene regulatory networks : a study on a biologically realistic dataset. *BMC Bioinformatics*, 9 :91, 2008.

[AKMM98a] Tatsuya Akutsu, Satoru Kuhara, Osamu Maruyama, and Satoru Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In *SODA '98 : Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 695–702, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[AKMM98b] Tatsuya Akutsu, Satoru Kuhara, Osamu Maruyama, and Satoru Miyano. A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions. *Genome Informatics*, 9 :151–160, 1998.

[AMK99] Tatsuya Akutsu, Satoru Miyano, and Satoru Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. *Pacific Symposium on Biocomputing*, pages 17–28, 1999.

[APS+02] Elena A. Ananko, Nikolay L. Podkolodny, Irina L. Stepanenko, Elena V. Ignatieva, Olga A. Podkolodnaya, and Nikolay A. Kolchanov. GeneNet : a database on structure and functional organisation of gene networks. *Nucleic Acids Research*, 30(1) :398–401, Jan 2002.

[ARM98] Adam Arkin, John Ross, and Harley H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells. *Genetics*, 149 :1633–1648, 1998.

[ASI02] Shin Ando, Erina Sakamoto, and Hitoshi Iba. Evolutionary modeling and inference of gene network. *Information Sciences — Informatics and Computer Science*, 145(3-4) :237–259, 2002.

[ATSB03] Constantin F. Aliferis, Ioannis Tsamardinos, Alexander R. Statnikov, and Laura E. Brown. Causal explorer : A causal probabilistic network learning toolkit for biomedical discovery. In Faramarz Valafar and Homayoun Valafar, editors, *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Scienes, METMBS '03, June 23-26, 2003, Las Vegas, Nevada, USA*, pages 371–376. CSREA Press, 2003.

[AW02] Vincent Auvray and Louis Wehenkel. On the construction of the inclusion boundary neighbourhood for markov equivalence classes of bayesian network structures. In *Uncertainty in Artificial Intelligence*, pages 26–35, 2002.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286 :509–512, Oct 1999.

[Bal94] Shumeet Baluja. Population-based incremental learning : A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1994.

[BBM93] David Beasley, David R. Bull, and Ralph R. Martin. An overview of genetic algorithms : Part 2, research topics. *University Computing*, 15(4) :170–181, 1993.

[BD97] Shumeet Baluja and Scott Davies. Combining multiple optimization runs with optimal dependency trees. Technical Report CMU-CS-97-157, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1997.

[BDY99] Amir Ben-Dor and Zohar Yakhini. Clustering gene expression patterns. In *Proceedings of the third annual international conference on Computational molecular biology (RECOMB 1999)*, pages 33–42, New York, NY, USA, 1999. ACM.

[BE05] Svetlana Bulashevska and Roland Eils. Inferring genetic regulatory logic from expression data. *Bioinformatics*, 21(11) :2706–2713, Jun 2005.

[BH95] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate : a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B*, 57 :289–300, 1995.

[BH05] Allister Bernard and Alexander J. Hartemink. Informative structure priors : joint learning of dynamic regulatory networks from multiple types of data. *Pacific Symposium on Biocomputing*, pages 459–470, 2005.

[Bic05] David R. Bickel. Probabilities of spurious connections in gene networks : application to expression time series. *Bioinformatics*, 21(7) :1121–1128, Apr 2005.

[BILn03] Rosa Blanco, Iñaki Inza, and Pedro Larrañaga. Learning Bayesian networks in the space of structures by estimation of distribution algorithms. *International Journal of Intelligent Systems*, 18(2) :205–220, 2003.

[BK99] Atul J. Butte and Isaac S. Kohane. Unsupervised knowledge discovery in medical databases using relevance networks. *Proceedings of the American Medical Informatics Association Symposium*, pages 711–715, 1999.

[BMS+05] Katia Basso, Adam A. Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Reverse engineering of regulatory networks in human b cells. *Nature Genetics*, 37(4) :382–390, Apr 2005.

[Boo85] Lashon B. Booker. Improving the performance of genetic algorithms in classifier systems. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 80–92, Mahwah, NJ, USA, 1985. Lawrence Erlbaum Associates, Inc.

[Bou93] Remco R. Bouckaert. Probabilistic network construction using the minimum description length principle. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty, European Conference (ECSQARU 1993)*, volume 747 of *Lecture Notes in Computer Science*, pages 41–48. Springer, 1993.

[BRCA00] Alvis Brazma, Alan Robinson, Graham Cameron, and Michael Ashburner. One-stop shop for microarray data. *Nature*, 403(6771) :699–700, Feb 2000.

[BSSS03] Vikram Budhraja, Edward Spitznagel, W. Timothy Schaiff, and Yoel Sadovsky. Incorporation of gene-specific variability improves expression analysis using high-density DNA microarrays. *BMC Biology*, 1 :1, 2003.

[BT98] Mark T. Borisuk and John J. Tyson. Bifurcation analysis of a model of mitotic control in frog eggs. *Journal of Theoretical Biology*, 195(1) :69–85, Nov 1998.

[BT00] Peter A. N. Bosman and Dirk Thierens. Continuous iterated density estimation evolutionary algorithms within the IDEA framework. In *Optimization By Building and Using Probabilistic*, pages 197–200, 2000.

[BTS+00] Atul J. Butte, Pablo Tamayo, Donna Slonim, Todd R. Golub, and Isaac S. Kohane. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences*, 97(22) :12182–12186, 2000.

[But00] John C. Butcher. *Numerical methods for ordinary differential equations in the 20th century*. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 2000.

[Can] SVM and kernel methods matlab toolbox. http ://asi.insa-rouen.fr/ arakotom/toolbox/index.html.

[CCNG+00] Katherine C. Chen, Attila Csikasz-Nagy, Bela Gyorffy, John Val, Bela Novak, and John J. Tyson. Kinetic analysis of a molecular model of the budding yeast cell cycle. *Molecular Biology of the Cell*, 11(1) :369–391, Jan 2000.

[CDD96] David Maxwell Chickering, Geiger Dan, and Heckermann David. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from data : AI and Statistics*, volume 5, pages 121–130, New York NY, 1996. Springer-Verlag.

[CGK+02] Jie Cheng, Russell Greiner, Jonathan Kelly, David Bell, and Weiru Liu. Learning bayesian networks from data : an information-theory based approach. *Artificial Intelligence*, 137(1–2)(1-2) :43–90, 2002.

[CH92] Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4) :309–347, 1992.

[CHC99] Ting Chen, Hongyu L. He, and George M. Church. Modeling gene expression with differential equations. In *Pacific Symposium on Biocomputing*, pages 29–40, 1999.

[Chi95] David Maxwell Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI 1995)*, pages 87–98, 1995.

[Chi02] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3 :507–554, 2002.

[Chu02] Gary A. Churchill. Fundamentals of experimental design for cDNA microarrays. *Nature Genetics*, 32 Suppl :490–495, Dec 2002.

[CL68] C.J.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3) :462–467, 1968.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill Book Company, 2001.

[CM02] Carlos Cotta and Jorge Muruzábal. Towards a more efficient evolutionary induction of Bayesian networks. In *PPSN VII : Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, pages 730–739, London, UK, 2002. Springer-Verlag.

[CM04] Carlos Cotta and Jorge Muruzábal. On the learning of Bayesian network graph structures via evolutionary programming. In P. Lucas, editor, *Proceedings of the Second Workshop on Probabilistic Graphical Models*, pages 65–72, Leiden, The Netherlands, 2004.

[CS95] Peter Cheeseman and John Stutz. *Bayesian classification (AutoClass) : theory and results*. Advances in Knowledge Discovery and Data Mining. AAAI Press/MIT Press, 1995.

[CT01] Carlos Cotta and José M. Troya. Analyzing directed acyclic graph recombination. In *Proceedings of the International Conference on Computational Intelligence, Theory and Applications, 7th Fuzzy Days*, volume 2206 of *Lecture Notes in Computer Science*, pages 739–748, London, UK, 2001. Springer.

[DBC06] Alain Delaplace, Thierry Brouard, and Hubert Cardot. Two evolutionary methods for learning bayesian network structures. In *Computational Intelligence and Security*, volume 1, pages 137 – 142, 2006.

[dBIJV97] Jeremy S. de Bonet, Charles Lee Isbell Jr., and Paul A. Viola. MIMIC : Finding optima by estimating probability densities. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, pages 424–430. The MIT Press, 1997.

[dBLP+05] Florence d'Alché Buc, Pierre-Jean Lahaye, Bruno-Édouard Perrin, Liva Ralaivola, Todor Vujasinovic, Aurélien Mazurie, and Samuele Bottani. *A dynamic model of gene regulatory networks based on inertia principle*, volume 176 of *Studies in Fuzziness and Soft Computing*, pages 93–117. Springer, 2005.

[DG89] Kalyanmoy Deb and David E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the third international conference on Genetic algorithms*, pages 42–50, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[dHIK+03] Michiel J.L. de Hoon, Seiya Imoto, Kazuo Kobayashi, Naotake Ogasawara, and Satoru Miyano. Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 8, pages 17–28, 2003.

[DJ75] Kenneth A. De Jong. *The analysis and behaviour of a class of genetic adaptative systems*. PhD thesis, University of Michigan, 1975.

[dJ02] Hidde de Jong. Modeling and simulation of genetic regulatory systems : a literature review. *Journal of Computational Biology*, 9(1)(1) :67–103, 2002.

[dJGHP03] Hidde de Jong, Johannes Geiselmann, Céline Hernandez, and Michel Page. Genetic network analyzer : qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3) :336–344, Feb 2003.

[dJP08] Hidde de Jong and Michel Page. Search for steady states of piecewise-linear differential equation models of genetic regulatory networks. *Transactions on Computational Biology and Bioinformatics*, 5(2) :208–222, 2008.

[dlFBHM04] Alberto de la Fuente, Nan Bing, Ina Hoeschele, and Pedro Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18) :3565–3574, 2004.

[DLS00] Patrik D'haeseleer, Shoudan Liang, and Roland Somogyi. Genetic network inference : from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8) :707–726, Aug 2000.

[DWFS98] Patrik D'haeseleer, Xiling Wen, Stefanie Fuhrman, and Roland Somogyi. Mining the gene expression matrix : inferring gene relationships from large scale gene expression data. In *IPCAT '97 : Proceedings of the second international workshop on Information processing in cell and tissues*, pages 203–212, New York, NY, USA, 1998. Plenum Press.

[DWFS99] Patrik D'haeseleer, Xiling Wen, Stefanie Fuhrman, and Roland Somogyi. Linear modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing*, pages 41–52, 1999.

[EBKR04] Claus Thorn Ekstrøm, Søren Bak, Charlotte Kristensen, and Mats Rudemo. Spot shape modelling and data transformations for microarrays. *Bioinformatics*, 20(14) :2270–2278, Sep 2004.

[ECS89] Larry J. Eshelman, Richard A. Caruana, and J. David Schaffer. Biases in the crossover landscape. In *Proceedings of the third international conference on Genetic algorithms*, pages 10–19, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[ELP97] Ramon Etxeberria, Pedro Larrañaga, and Juan M. Picaza. Analysis of the behaviour of genetic algorithms when learning Bayesian network structure from data. *Pattern Recognition Letters*, 18(11-13) :1269–1273, 1997.

[ER59] Paul Erdös and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6 :290–297, 1959.

[ESBB98] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25) :14863–14868, Dec 1998.

[ETST01] Bradley Efron, Robert Tibshirani, John D. Storey, and Virginia Tusher. Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96 :1151–1160, December 2001.

[FGA99] Nir Friedman, Moises Goldszmidt, and Wyner Abraham. Data analysis with Bayesian networks : a bootstrap approach. *Uncertainty in Artificial Intelligence : Proceedings of the 15'th Conference*, 1999.

[FK03] Nir Friedman and Daphne Koller. Being Bayesian about network structure : A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50 :95–126, 2003.

[FL04] Olivier François and Philippe Leray. étude comparative d'algorithmes d'apprentissage de structure dans les réseaux Bayésiens. *Journal électronique d'intelligence artificielle*, 5(39) :1–19, 2004.

[FNCT06] Adrien Fauré, Aurélien Naldi, Claudine Chaouiya, and Denis Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. In *Proceedings 14th International Conference on Intelligent Systems for Molecular Biology (Supplement of Bioinformatics)*, pages 124–131, Jul 2006.

[FNP99] Nir Friedman, Iftach Nachman, and Dana Pe'er. Learning bayesian network structure from massive datasets : The "sparse candidate" algorithm. In *Uncertainty in Artificial Intelligence*, pages 206–215, 1999.

[Fra06] Olivier François. *De l'identification de structure de réseaux bayésiens à la reconnaissance de formes à partir d'informations complètes ou incomplètes*. PhD thesis, french National Institute of Applied Sciences (INSA) of Rouen, 2006.

[GB98] Michael Gibson and Jehoshua Bruck. An efficient algorithm for generating trajectories of stochastic gene regulation reactions. Technical Report ETR026, California Institute of Technology, 1998.

[GDH92] David E. Goldberg, Kalyanmoy Deb, and Jeffrey Horn. Massive multimodality, deception, and genetic algorithms. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, Amsterdam, 1992. Elsevier Science Publishers, B. V.

[GDR03] Thomas Gärtner, Kurt Driessens, and Jan Ramon. Graph kernels and gaussian processes for relational reinforcement learning. In *Thirteenth International Conference on Inductive Logic Programming (ILP-2003)*. Springer, 2003.

[GH97] Dan Geiger and David Heckerman. A characterization of the Dirichlet distribution through global and local independence. *The Annals of Statistics*, 25(3) :1344–1369, 1997.

[GHS99] Igor Goryanin, T. Charles Hodgman, and Evgeni Selkov. Mathematical simulation and analysis of cellular metabolism and regulation. *Bioinformatics*, 15(9) :749–758, Sep 1999.

[Gil77] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81 :2340–2361, 1977.

[Gil92] Daniel T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A, Statistical Mechanics and its Applications*, 188 :404–425, September 1992.

[GK72] L. Glass and S. A. Kauffman. Co-operative components, spatial localization and oscillatory cellular dynamics. *Journal of Theoretical Biology*, 34(2) :219–237, Feb 1972.

[GK73] L. Glass and S. A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology*, 39(1) :103–129, Apr 1973.

[Gla75] L. Glass. Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology*, 54(1) :85–107, Oct 1975.

[Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc., 1989.

[GPS+05] Nikhil R. Garge, Grier P. Page, Alan P. Sprague, Bernard S. Gorman, and David B. Allison. Reproducible clusters from microarray research : Whither ? *BMC Bioinformatics*, 6 Suppl 2 :S10, Jul 2005.

[GR87] David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 41–49, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.

[Har95] Georges R. Harik. Finding multimodal solutions using restricted tournament selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 24–31, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[Har99] Georges R. Harik. Linkage learning via probabilistic modeling in the ECGA. Technical Report 99010, University of Illinois, 1999.

[Has70] W. Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1) :97 – 109, 1970.

[HG95] David Heckerman and Dan Geiger. Likelihoods and parameter priors for Bayesian networks. Technical Report MSR-TR-95-54, MicroSoft Research, 1995.

[HGC95] David Heckerman, Dan Geiger, and David Maxwell Chickering. Learning Bayesian networks : The combination of knowledge and statistical data. *Machine Learning*, 20(3) :197–243, 1995.

[HKK07] Desmond J. Higham, Gabriela Kalna, and Milla Kibble. Spectral clustering and its use in bioinformatics. *Journal of Computational and Applied Mathematics*, 204(1) :25–37, 2007.

[HLG99] Georges R. Harik, Fernando G. Lobo, and David E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4) :287–297, 1999.

[Hol75] John H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[HSKG99] C. Hill, B. Sawhill, S. Kauffman, and L. Glass. Transition to chaos in models of genetic networks. In Springer, editor, *Statistical Mechanics of Biocomplexity*, volume 527 of *Lecture Notes in Physics*, pages 261–274, 1999.

[HSL+02] Jianjun Hu, Kisung Seo, Shaobo Li, Zhun Fan, Ronald C. Rosenberg, and Erik D. Goodman. Structure fitness sharing (SFS) for evolutionary design by genetic programming. In W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, editors, *GECCO 2002 : Proceedings of the Genetic and Evolutionary Computation Conference*, pages 780–787, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[HSSS02] Daehee Hwang, William A Schmitt, George Stephanopoulos, and Gregory Stephanopoulos. Determination of minimum sample size and discriminatory expression patterns in microarray data. *Bioinformatics*, 18(9) :1184–1193, Sep 2002.

[Hua99] S. Huang. Gene expression profiling, genetic networks, and cellular states : an integrating concept for tumorigenesis and drug discovery. *J Mol Med*, 77(6) :469–480, Jun 1999.

[Hus03] Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19 :2271–2282, 2003.

[IGM02] Seiya Imoto, Takao Goto, and Satoru Miyano. Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. *Pac Symp Biocomput*, pages 175–186, 2002.

[IKG+02] Seiya Imoto, SunYong Kim, Takao Goto, Sachiyo Aburatani, Kousuke Tashiro, Satoru Kuhara, and Satoru Miyano. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. In *IEEE Computer Society Bioinformatics Conference*, pages 219–227, 2002.

[IKG+03] Seiya Imoto, Sunyong Kim, Takao Goto, Satoru Miyano, Sachiyo Aburatani, Kousuke Tashiro, and Satoru Kuhara. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *J Bioinform Comput Biol*, 1(2) :231–252, Jul 2003.

[ITK00] T. E. Ideker, V. Thorsson, and R. M. Karp. Discovery of regulatory interactions through perturbation : inference and experimental design. *Pac Symp Biocomput*, pages 305–316, 2000.

[IWJ06] Rafael A Irizarry, Zhijin Wu, and Harris A Jaffee. Comparison of Affymetrix GeneChip expression measures. *Bioinformatics*, 22(7) :789–794, Apr 2006.

[JGH+04] Hidde De Jong, Jean-Luc Gouzé, Céline Hernandez, Michel Page, Tewfik Sari, and Johannes Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull Math Biol*, 66(2) :301–340, Mar 2004.

[JN06] Martin Janvraisemblancezura and Jan Nielsen. A simulated annealing-based method for learning Bayesian networks from statistical data : Research articles. *Int. J. Intell. Syst.*, 21(3) :335–348, 2006.

[JS91] Kenneth A. De Jong and William M. Spears. An analysis of the interacting roles of population size and crossover in genetic algorithms. In *PPSN I : Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, pages 38–47. Springer-Verlag, 1991.

[KAG+08] Minoru Kanehisa, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, and Yoshihiro Yamanishi. Kegg for linking genomes to life and the environment. *Nucleic Acids Res*, 36(Database issue) :D480–D484, Jan 2008.

[Kam07] N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry (3rd edition)*. 2007.

[Kau69] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol*, 22(3) :437–467, Mar 1969.

[Kau74] S. Kauffman. The large scale structure and dynamics of gene control circuits : an ensemble approach. *J Theor Biol*, 44(1) :167–190, Mar 1974.

[Kau93] Stuart A. Kauffman. *The Origins of Order : Self-Organization and Selection in Evolution*. Oxford University Press, May 1993.

[KC01a] M. K. Kerr and G. A. Churchill. Experimental design for gene expression microarrays. *Biostatistics*, 2(2) :183–201, Jun 2001.

[KC01b] Tomas Kocka and Robert Castelo. Improved learning of Bayesian networks. In *UAI '01 : Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 269–276, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[KEG03] K. Kappler, R. Edwards, and L. Glass. Dynamics in high-dimensional model gene networks. *Signal Process.*, 83(4) :789–798, 2003.

[KL51] Solomon Kullback and Richard Liebler. On information and sufficiency. *Annals of Mathematical Statistics*, 22 :79–86, 1951.

[KLP07] Haseong Kim, Jae K Lee, and Taesung Park. Boolean networks using the chi-square test for inferring large-scale gene regulatory networks. *BMC Bioinformatics*, 8 :37, 2007.

[KPP95] Vladimir Kvasnicka, Martin Pelikan, and Jiri Pospichal. Hill-climbing with learning : An abstraction of genetic algorithm. Technical report, Slovak Technical University, 1995.

[KW00] H. Kishino and P. J. Waddell. Correspondence analysis of genes and tissue types and finding genetic links from microarray data. *Genome Inform Ser Workshop Genome Inform*, 11 :83–95, 2000.

[LB94] Wai Lam and Fahiem Bacchus. Learning bayesian belief networks : An approach based on the MDL principle. *Computational Intelligence*, 10(4) :269–293, 1994.

[LBU04] Phillip P. Le, Amit Bahl, and Lyle H. Ungar. Using prior knowledge to improve genetic network reconstruction from microarray data. *In Silico Biology*, 4(2), 2004.

[Ler] Philippe Leray. Bnt structure learning package. http ://bnt.insa-rouen.fr/.

[Ler06] Philippe Leray. *Réseaux Bayésiens - Apprentissage et modélisation de systémes complexes*. Habilitation à diriger des recherches, Université de Rouen, 2006.

[LFS98] S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput*, pages 18–29, 1998.

[LHY05] Xiao-Lin Li, Xiang-Dong He, and Sen-Miao Yuan. Learning Bayesian networks structures from incomplete data based on extending evolutionary programming. In *Machine Learning and Cybernetics*, volume 4, pages 2039– 2043, August 2005.

[LnKMY96] Pedro Larrañaga, Cindy M. H. Kuijpers, Roberto H. Murga, and Yosu Yurramendi. Learning Bayesian network structures by searching for the bestordering with genetic algorithms. *Systems, Man and Cybernetics, Part A, IEEE Transactions*, 26(4) :487 – 493, 1996.

[LnPY+96] Pedro Larrañaga, Mikel Poza, Yosu Yurramendi, Roberto H. Murga, and Cindy M. H. Kuijpers. Structure learning of Bayesian networks by genetic algorithms : A performance analysis of control parameters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(9) :912–926, 1996.

[LSYH03] Harri Lähdesmäki, Ilya Shmulevich, and Olli Yli-Harja. On learning gene regulatory networks under the boolean network model. *Machine Learning*, 52(1-2) :147–167, 2003.

[MA97] H. H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *Proc Natl Acad Sci U S A*, 94(3) :814–819, Feb 1997.

[Mah95] Samir W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1995.

[MC04] Jorge Muruzábal and Carlos Cotta. A primer on the evolution of equivalence classes of Bayesian-network structures. In *Parallel Problem Solving from Nature - PPSN VIII, 8th International Conference*, pages 612–621, 2004.

[MCA+98] G. S. Michaels, D. B. Carr, M. Askenazi, S. Fuhrman, X. Wen, and R. Somogyi. Cluster analysis and data visualization of large-scale gene expression data. *Pac Symp Biocomput*, pages 42–53, 1998.

[Men93] P. Mendes. Gepasi : a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Comput Appl Biosci*, 9(5) :563–571, Oct 1993.

[MK04] Paul M Magwene and Junhyong Kim. Estimating genomic coexpression networks using first-order conditional independence. *Genome Biol*, 5(12) :R100, 2004.

[MLD99] James W. Myers, Kathryn B. Laskey, and Kenneth A. DeJong. Learning Bayesian networks from incomplete data using evolutionary algorithms. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 458–465, Orlando, Florida, USA, 13-17 1999. Morgan Kaufmann.

[MM77] Leonard A. Marascuilo and Maryellen McSweeney. *Nonparametric and distribution-free methods for the social sciences*. Brooks/Cole, 1977.

[MM99] Heinz Mühlenbein and Thilo Mahnig. FDA - A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4) :353–376, 1999.

[MMB03] Carmen G Moles, Pedro Mendes, and Julio R Banga. Parameter estimation in biochemical pathways : a comparison of global optimization methods. *Genome Res*, 13(11) :2467–2474, Nov 2003.

[MNB+06] A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, D. Favera, and A. Califano. ARACNE : An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 2 :S1–7, 2006.

[MP96] Heinz Mühlenbein and Gerhard Paass. From recombination of genes to the estimation of distributions i. binary parameters. In *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature (PPSN 1996)*, volume 1141 of *Lecture Notes in Computer Science*, pages 178–187. Springer, 1996.

[MPO95] Thomas Mestl, Erik Plahte, and Stig W. Omholt. A mathematical framework for describing and analysing gene regulatory networks. *Journal of Theoretical Biology*, 176 :291–300, 1995.

[MS95] H. H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269(5224) :650–656, Aug 1995.

[MS07] Florian Markowetz and Rainer Spang. Inferring cellular networks–a review. *BMC Bioinformatics*, 8 Suppl 6 :S5, 2007.

[MSY03] Pedro Mendes, Wei Sha, and Keying Ye. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19 Suppl 2 :II122–II129, Oct 2003.

[MTR+03] Sayan Mukherjee, Pablo Tamayo, Simon Rogers, Ryan Rifkin, Anna Engle, Colin Campbell, Todd R Golub, and Jill P Mesirov. Estimating dataset size requirements for classifying DNA microarray data. *J Comput Biol*, 10(2) :119–142, 2003.

[Müh97] Heinz Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3) :303–346, 1997.

[Mur] Kevin Murphy. Bayes net toolbox (http ://bnt.sourceforce.net/).

[Mur02] Kevin Murphy. *Dynamic Bayesian Networks : Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, 2002.

[NRF04] Iftach Nachman, Aviv Regev, and Nir Friedman. Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20(1) :248–256, Aug 2004.

[NST+98] Noda, Shinohara, Takeda, Matsumoto, Miyano, and Kuhara. Finding genetic network from experiments by weighted network model. *Genome Inform Ser Workshop Genome Inform*, 9 :141–150, 1998.

[NT95] Bela Novak and John J. Tyson. Quantitative analysis of a molecular model of mitotic control in fission yeast. *Journal of Theoretical Biology*, 173 :283–305, 1995.

[PDR08] F. Picard, J.-J. Daudin, and S. Robin. A mixture model for random graphs. *Stat. Comput.*, 18(2) :173–83, 2008.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[Pea00] Judea Pearl. *Causality : Models, Reasoning, and Inference*. Cambridge University Press, March 2000.

[PGCP99] Martin Pelikan, David E. Goldberg, and Erick Cantú-Paz. BOA : The Bayesian optimization algorithm. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO 1999*, volume 1, pages 525–532, 1999.

[PM99] Martin Pelikan and Heinz Mühlenbein. The bivariate marginal distribution algorithm. In R. Roy, T. Furuhashi, and P. K. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535. Springer, 1999.

[PMO95] Erik Plahte, Thomas Mestl, and Stig W. Omholt. Stationary states in food web models with threshold relationships. *Journal of Biological Systems*, 3 :569 – 577, 1995.

[PMO98] E. Plahte, T. Mestl, and S. W. Omholt. A methodological basis for description and analysis of systems with complex switch-like interactions. *J Math Biol*, 36(4) :321–348, Mar 1998.

[PnBT05] José M. Peña, Johan Bjorkegren, and Jesper Tegnér. Growing Bayesian network models of gene networks from seed genes. *Bioinformatics*, 21 Suppl 2 :ii224–ii229, Sep 2005.

[PnLLn04] José M. Peña, José Antonio Lozano, and Pedro Larrañaga. Unsupervised learning of Bayesian networks via estimation of distribution algorithms : an application to gene expression data clustering. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(Supplement-1) :63–82, 2004.

[PRM+03] Bruno-Édouard Perrin, Liva Ralaivola, Aurélien Mazurie, Samuele Bottani, Jacques Mallet, and Florence d'Alché Buc. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19 :138–148, 2003.

[PSS00] Clark Glymour Peter Spirtes and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, 2000.

[QBdB07] Minh Quach, Nicolas Brunel, and Florence d'Alché Buc. Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, 23 :3209 – 3216, 2007.

[Qua02] John Quackenbush. Microarray data normalization and transformation. *Nat Genet*, 32 Suppl :496–501, Dec 2002.

[Raf95] A. Raftery. Bayesian model selection in social research. *Sociological Methodology*, 1995. with discussion by Andrew Gelman, Donald B. Rubin, and Robert M. Hauser, and a rejoinder.

[RBB06] David J Reiss, Nitin S Baliga, and Richard Bonneau. Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. *BMC Bioinformatics*, 7 :280, 2006.

[RC03] Claudine Robert and Olivier Cogis. *Théorie des Graphes*. Vuibert, 2003.

[RC07] Adrien Richard and Jean-Paul Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Appl. Math.*, 155(18) :2403–2413, 2007.

[RHCC07] J.O. Ramsay, G. Hooker, J. Cao, and D. Campbell. Parameter estimation for differential equations : A generalized smoothing approach. *Journal of the Royal Statistical Society (B)*, 69 :741–796, 2007.

[Ris78] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14 :445–471, 1978.

[RMS95] J. Reinitz, E. Mjolsness, and D. H. Sharp. Model for cooperative control of positional information in drosophila by bicoid and maternal hunchback. *J Exp Zool*, 271(1) :47–56, Jan 1995.

[Rob77] R.W. Robinson. Counting unlabeled acyclic digraphs. *Lecture Notes in Mathematics*, 622 :239–273, 1977.

[RPKL07] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p*) models for social networks. *Social Networks*, 29 :173–191, 2007.

[RRW+00] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young. Genome-wide location and function of DNA binding proteins. *Science*, 290(5500) :2306–2309, Dec 2000.

[RV90] J. Reinitz and J. R. Vaisnys. Theoretical and experimental analysis of the phage lambda genetic switch implies missing levels of co-operativity. *J Theor Biol*, 145(3) :295–318, Aug 1990.

[Sam69] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5) :401–409, 1969.

[SBA07] N. Soranzo, G. Bianconi, and C. Altafini. Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks : synthetic versus real data. *Bioinformatics*, 24(13) :1640–1647, 2007.

[Sch78] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2) :461–464, 1978.

[SD91] William M. Spears and Kenneth A. De Jong. An analysis of multi-point crossover. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 301–315, San Mateo, CA, 1991. Morgan Kaufmann.

[SDKZ02] Ilya Shmulevich, Edward R Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic boolean networks : a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2) :261–274, Feb 2002.

[SDZ02a] Ilya Shmulevich, Edward R. Dougherty, and Wei Zhang. From boolean to probabilistic boolean networks as models of genetic regulatory networks. *IEEE*, 90(11) :1778–1792, 2002.

[SDZ02b] Ilya Shmulevich, Edward R Dougherty, and Wei Zhang. Gene perturbation and intervention in probabilistic boolean networks. *Bioinformatics*, 18(10) :1319–1331, Oct 2002.

[SF63] M. Sugita and N. Fukuda. Functional analysis of chemical systems in vivo using a logical circuit equivalent. 3. analysis using a digital circuit combined with an analogue computer. *J Theor Biol*, 5(3) :412–425, Nov 1963.

[SGH+03] Ilya Shmulevich, Ilya Gluhovsky, Ronaldo F. Hashimoto, E.R. Dougherty, and Zhank Wei. Steady-state analysis of genetic regulatory networks modelled by probabilistic boolean networks. *Comparative and functional genomics*, 4 :601–608, 2003.

[SGS+00] Peter Spirtes, Clark Glymour, Richard Scheines, Stuart Kauffman, Valerio Aimale, and Frank Wimberly. Constructing Bayesian network models of gene expression networks from microarray data. In *Atlantic Symposium on Computational Biology, Genome Information Systems & Technology.*, 2000.

[Sha48] C. E. Shannon. A mathematical theory of communication. *The Bell System technical journal*, 27 :379–423, 1948.

[SJH02] V. Smith, E. Jarvis, and A. Hartemink. Evaluating functional network inference using simulations of complex biological systems, 2002.

[SK98] B. Sareni and L. Krahenbuhl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3) :97–106, 1998.

[SKFW03a] R. Steuer, J. Kurths, O. Fiehn, and W. Weckwerth. Interpreting correlations in metabolomic networks. *Biochem Soc Trans*, 31(Pt 6) :1476–1478, Dec 2003.

[SKFW03b] R. Steuer, J. Kurths, O. Fiehn, and W. Weckwerth. Observing and interpreting correlations in metabolomic networks. *Bioinformatics*, 19(8) :1019–1026, May 2003.

[SKPG05] Hana El Samad, Mustafa Khammash, Linda Petzold, and Dan Gillespie. Stochastic modelling of gene regulatory networks. *International Journal of Robust and Nonlinear Control*, 15 :691 – 711, 2005.

[SMF07] Anthony Martino Shawn Martin, Zhaoduo Zhang and Jean-Loup Faulon. Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics*, 23 :866–874, 2007.

[SR98] D. H. Sharp and J. Reinitz. Prediction of mutant expression patterns using gene circuits. *Biosystems*, 47(1-2) :79–90, 1998.

[SS05] J. Schafer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4 :32, 2005.

[SSM97] B. Schölkopf, A.J. Smola, and K.R. Müller. Kernel principal component analysis. In M. Hasler W. Gerstner, A. Germond and J.-D. Nicoud, editors, *7th International Conference on Artificial Neural Networks, ICANN 97, Lausanne, Switzerland*, volume 1327, pages 583–588, Berlin, 1997. Springer Lecture Notes in Computer Science.

[SSR+03] Eran Segal, Michael Shapira, Aviv Regev, Dana Pe'er, David Botstein, Daphne Koller, and Nir Friedman. Module networks : identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet*, 34(2) :166–76, Jun 2003.

[SSZ+98] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9(12) :3273–3297, Dec 1998.

[ST92] Houssine El Snoussi and Rene Thomas. Logical identification of all steady states : The concept of feedback loop characteristic states. *Bulletin of Mathematical Biology*, 55 :973–991, 1992.

[STM+05] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, and Jill P Mesirov. Gene set enrichment analysis : a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, 102(43) :15545–15550, Oct 2005.

[Str00] Steven Strogatz. *Nonlinear Dynamics and Chaos : With Applications to Physics, Biology, Chemistry, and Engineering*. Perseus Books, 2000.

[SWS+01] M. Steinfath, W. Wruck, H. Seidel, H. Lehrach, U. Radelof, and J. O'Brien. Automated image analysis for array hybridization experiments. *Bioinformatics*, 17(7) :634–641, Jul 2001.

[Sys89] Gilbert Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 2–9, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[SYS03] Gordon K Smyth, Yee Hwa Yang, and Terry Speed. Statistical issues in cDNA microarray data analysis. *Methods Mol Biol*, 224 :111–136, 2003.

[Sza99] Zoltan Szallasi. Genetic network analysis in light of massively parallel biological data acquisition. In *Pacific Symposium on Biocomputing*, pages 5–16, 1999.

[TBA06] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1) :31–78, 2006.

[TCM04] Pawan Dhar Tan Chee Meng, Sandeep Somani. Modeling and simulation of biological systems with stochasticity. *In Silico Biology*, 4 :293–309, 2004.

[TGGF08] Arthur Tenenhaus, Vincent Guillemot, Xavier Gidrol, and Vincent Frouin. Gene association networks from microarray data using a regularized estimation of partial correlation based on PLS regression. *soumis à IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2008.

[TH02] Hiroyuki Toh and Katsuhisa Horimoto. Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, 18(2) :287–297, Feb 2002.

[Tho73] R. Thomas. Boolean formalization of genetic control circuits. *J Theor Biol*, 42(3) :563–585, Dec 1973.

[Tho91] R. Thomas. Regulatory networks seen as asynchronous automata : a logical description. *Journal of theoretical biology*, 153 :1–23, 1991.

[Tho98] R. Thomas. Laws for the dynamics of regulatory networks. *Int J Dev Biol*, 42(3) :479–485, 1998.

[TKB+03] Yoshinori Tamada, SunYong Kim, Hideo Bannai, Seiya Imoto, Kousuke Tashiro, Satoru Kuhara, and Satoru Miyano. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. In *ECCB*, pages 227–236, 2003.

[TTC01] V. G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*, 98(9) :5116–5121, Apr 2001.

[TTK95] R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviour of biological regulatory networks–i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull Math Biol*, 57(2) :247–276, Mar 1995.

[TW05] George C Tseng and Wing H Wong. Tight clustering : a resampling-based approach for identifying stable and tight patterns in data. *Biometrics*, 61(1) :10–16, Mar 2005.

[VP91] Tom Verma and Judea Pearl. *Equivalence and Synthesis of Causal Models*. Elsevier Science, New York, NY, 1991.

[vRS06] N. A W van Riel and E. D. Sontag. Parameter estimation in models combining signal transduction and metabolic pathways : the dependent input approach. *Syst Biol (Stevenage)*, 153(4) :263–274, Jul 2006.

[vSWR00] E. P. van Someren, L. F. Wessels, and M. J. Reinders. Linear modeling of genetic networks from experimental data. *Proc Int Conf Intell Syst Mol Biol*, 8 :355–366, 2000.

[WB06] A. Wille and P. Bühlmann. Low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, 5 :issue 1, article 1, 2006.

[WGH06] Adriano V Werhli, Marco Grzegorczyk, and Dirk Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and Bayesian networks. *Bioinformatics*, 22(20) :2523–2531, Oct 2006.

[WK00] P. J. Waddell and H. Kishino. Cluster inference methods and graphical models evaluated on NCI60 microarray gene expression data. *Genome Inform Ser Workshop Genome Inform*, 11 :129–140, 2000.

[WKB05] Cecily J Wolfe, Isaac S Kohane, and Atul J Butte. Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. *BMC Bioinformatics*, 6 :227, 2005.

[WLL99] M. L. Wong, W. Lam, and K. S. Leung. Using evolutionary programming and minimum description length principle for data mining of Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2) :174–178, 1999.

[WMH03] Junbai Wang, Ola Myklebost, and Eivind Hovig. MGraph : graphical models for microarray data analysis. *Bioinformatics*, 19(17) :2210–2211, Nov 2003.

[WP00] RA Watson and JB Pollack. Combination and recombination in genetic algorithms. Technical Report CS-00-209, Dept. Computer Science, Brandeis University, 2000.

[WPY67] C. Walter, R. Parker, and M. Ycas. A model for binary logic in biochemical systems. *J Theor Biol*, 15(2) :208–217, May 1967.

[WTX04] Tie Wang, J.W. Touchman, and Guoliang Xue. Applying two-level simulated annealing on Bayesian structure learning to infer genetic networks. In *Proc. IEEE Computational Systems Bioinformatics Conference CSB 2004*, pages 647–648, 2004.

[WWS99] D. C. Weaver, Christopher T. Workman, and Gary D. Stormo. Modeling regulatory networks with weight matrices. In *Pacific Symposium on Biocomputing*, pages 112–123, 1999.

[YBDS00] Y. Yang, M. Buckley, S. Dudoit, and T. Speed. Comparison of methods for image analysis on cdna microarray data, 2000.

[YDL+02] Yee Hwa Yang, Sandrine Dudoit, Percy Luu, David M Lin, Vivian Peng, John Ngai, and Terence P Speed. Normalization for cDNA microarray data : a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res*, 30(4) :e15, Feb 2002.

[YZO+06] N. Yalamanchili, D. E. Zak, B. A. Ogunnaike, J. S. Schwaber, A. Kriete, and B. N. Kholodenko. Quantifying gene network connectivity in silico : scalability and accuracy of a modular approach. *Syst Biol (Stevenage)*, 153(4) :236–246, Jul 2006.

[ZZ00] K. Zhang and H. Zhao. Assessing reliability of gene clusters from gene expression data. *Funct Integr Genomics*, 1(3) :156–173, Nov 2000.

Table des matières