Apprentissage de la structure de réseaux bayésiens. Application aux données de génétique-génomique

Jimmy Vandel

▶ **To cite this version:**

Jimmy Vandel. Apprentissage de la structure de réseaux bayésiens. Application aux données de génétique-génomique. Mathématiques [math]. Université Toulouse III - Paul Sabatier, 2012. Français. NNT : . tel-02809699

HAL Id: tel-02809699
https://hal.inrae.fr/tel-02809699

Submitted on 6 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Discipline : *Informatique – Intelligence Artificielle*

Présentée et soutenue par *Jimmy VANDEL*
Le 7 Décembre 2012

Titre : *Apprentissage de la structure de réseaux bayésiens. Application aux données de génétique-génomique.*

JURY

Mr Gilles RICHARD	Professeur des Universités - Université de Toulouse	*Président*
Mme Florence D'ALCHÉ-BUC	Professeur des Universités - Université d'Évry	*Rapporteur*
Mr Philippe LERAY	Professeur des Universités - Université de Nantes	*Rapporteur*
Mme Sylvie HUET	Directeur de Recherche - INRA Jouy-en-Josas	*Examinateur*
Mr Sébastien AUBOURG	Directeur de Recherche - INRA Évry	*Examinateur*
Mme Brigitte MANGIN	Directeur de Recherche - INRA Toulouse	*Directeur*
Mr Simon DE GIVRY	Chargé de Recherche - INRA Toulouse	*Co-directeur*

École doctorale : *Mathématiques, Informatique,Télécommunications de Toulouse (MITT)*
Unité de recherche : *INRA, Unité de Biométrie et d'Intelligence Artificielle, UR875, Toulouse*
Directeurs de Thèse : *Mme Brigitte MANGIN et Mr Simon DE GIVRY*
Rapporteurs : *Mme Florence D'ALCHÉ-BUC et Mr Philippe LERAY*


SIGILLVM·SCOLARIVM·TOLOSE·
VNIVERSITATIS·MAGISTRORVM·ET·SCOLARIVM

Remerciements

Avant de vous laisser à la lecture de ce manuscrit, je me livre bien volontiers à la délivrance de quelques sincères remerciements aux personnes ayant contribué de manière plus ou moins directe à cette thèse.

Je tiens, dans un premier temps, à remercier les différents membres du jury en commençant par Florence d'Alché-Buc et Philippe Leray qui ont gentiment accepté de rapporter cette thèse et dont les conseils ont permis d'en améliorer le contenu. Je remercie également Sylvie Huet et Sébastien Aubourg qui, en dépit d'une météo capricieuse et d'une visio récalcitrante, ont pointé des perspectives intéressantes à ce travail et enfin Gilles Richard pour son regard extérieur toujours bénéfique.

Également présents dans ce jury mais surtout tout au long de cette thèse, je remercie Brigitte et Simon en leur adressant des félicitations, car cette thèse ne serait pas ce qu'elle est sans ces 3 années de réunions et de discussions toujours productives. En dépit de la distance qui sépare encore le monde des statistiques de celui de l'informatique, nous avons réussi au final à trouver un langage commun basé sur la compréhension de chacun et la bonne humeur. Toujours disponibles tout en me laissant choisir les voies à explorer, c'est avec un réel plaisir que j'ai passé ces 3 années à vos cotés.

J'adresse aussi mes remerciements à l'ensemble des membres de l'unité BIA pour leur gentillesse et leur contribution à la vie de l'unité, rendant le quotidien du thésard bien plus agréable. Je ne peux malheureusement pas citer tout le monde alors je me contenterai de faire quelques mentions particulières. Je commence donc par nos secrétaires actuelles Nathalie et Fabienne sans oublier leurs prédécesseurs, Pascale et Jacky, indispensables au fonctionnement du labo tout comme le sont Mickaël et Abde qui ont empêché maintes défenestrations d'écran lors de soucis informatiques. Je remercie également Régis qui en tant que directeur d'unité, a su faire passer ~~le rugby~~... euh l'unité avant tout et qui nous a autorisés à rester travailler très tard le soir ! Mention spéciale aux autres thésards de l'unité qui ont déjà soutenu ou qui le feront dans un avenir proche. Pour les anciens je remercie Aurélie, Éric et Mahuna, qui ont su nous mettre dans le bain dès notre arrivée et nous ont montré la voie à suivre. Je n'oublie pas les thésards plus récents comme Hiep, Magali et Julia à qui je souhaite encore beaucoup de plaisir au BIA. Je remercie également Ronan, Gauthier, Éric et Pierre pour les repas partagés ensemble et Jérôme pour m'avoir fait découvrir les joies des coteaux.

Si j'utilise parfois dans ces remerciements, le pronom "nous" plutôt que "je" c'est que j'ai eu la chance d'avoir, durant ces 3 années, un indéfectible compagnon d'infortune à qui je dois beaucoup et qui est, de fait, un peu responsable de ce manuscrit. Pour ces nombreuses heures passées à commenter l'actualité, à débattre, à commérer comme deux petites vieilles autour d'un thé, Mathieu je t'adresse, plutôt qu'un simple merci, une gros poutou avec toute mon amitié.

Parler de Mathieu mène inévitablement à aborder le cas de son homonyme situé dans le bureau voisin et qui vient compléter ce triumvirat de gentils compagnons. Matthieu, entre ton style vestimentaire inclassable et ta volonté de faire partager tes goûts musicaux à toute l'unité, tu mérites amplement ton grade de grand-gamin en chef. Ainsi, moi et Mathieu, tes fidèles disciples, te respecterons éternellement pour cela (et uniquement pour cela). Une bise pour toi mais aussi à ta petite famille, Marilyne, Tristan et Charlotte sans oublier le pauvre Vengo.

Parce qu'une thèse ne se résume pas qu'à la seule vie au labo, je remercie également toutes les personnes rencontrées à Toulouse avec qui il n'est malheureusement pas toujours évident de garder contact, mais à qui je pense chaleureusement. Petites pensées particulières à Armelle, Sarah et Vincent avec qui j'ai passé d'excellentes soirées au "Bureau" ou lors de mes très rares sorties au Shanghai. Pensées plus affectueuses cette fois, pour les trois loustiques qui m'ont accompagné durant ces années toulousaines à savoir Messieurs Benoit, Olivier et Enguerran que j'embrasse fort et que je remercie pour tout ce qu'ils m'ont apporté.

Et enfin je réserve mes dernières pensées pour mes parents, qui représentent depuis toutes ces années universitaires une source de motivation à demi-avouée et pour ma grand-mère, qui je l'espère, aurait été fière de son petit-fils.

Résumé

Apprendre la structure d'un réseau de régulation de gènes est une tâche complexe due à la fois au nombre élevé de variables le composant (plusieurs milliers) et à la faible quantité d'échantillons disponibles (quelques centaines). Parmi les approches proposées permettant d'apprendre ces réseaux, nous nous sommes placés pour cette thèse dans le formalisme des réseaux bayésiens. Apprendre la structure d'un réseau de régulation consiste alors à apprendre la structure d'un réseau bayésien où chaque variable représente un gène et chaque arc de ce réseau un phénomène de régulation.

Dans la première partie de cette thèse nous nous intéressons à l'apprentissage de la structure de réseaux bayésiens génériques. Nous nous sommes placés dans le cadre des recherches locales à base de score. Nous proposons une version stochastique d'une recherche gloutonne existante, permettant d'explorer plus efficacement l'espace des réseaux possibles. Pour ce même objectif, nous avons développé un nouvel opérateur local, le SWAP, ainsi qu'une extension itérative des opérateurs classiques, permettant d'assouplir temporairement la contrainte d'acyclicité imposée par le formalisme des réseaux bayésiens.

La deuxième partie vise plus spécifiquement l'apprentissage de réseaux de régulation de gènes. Nous proposons une modélisation de ce problème d'apprentissage dans le cadre des réseaux bayésiens qui permet la prise en compte de deux types d'information. Le premier, classiquement utilisé, est le niveau d'expression des différents gènes. Le second, plus original, est la présence de mutations sur la séquence d'ADN pouvant expliquer certaines variations de l'expression. L'utilisation de ce type de données dites de *génétique-génomique*, vise à améliorer la qualité du réseau reconstruit en intégrant différentes sources d'information lors de l'apprentissage. Nous avons développé deux variantes de cette modélisation dont la première consiste en une représentation non-fusionnée des deux informations augmentant alors la qualité descriptive du réseau tandis que la seconde permet une représentation compacte. Nous avons également défini une extension des scores classiquement employés pour l'apprentissage, permettant de restaurer un a priori uniforme sur les classes de connectivité des réseaux explorés. Les deux modélisations proposées ainsi que l'utilisation des scores étendus ont été validées sur des données simulées issues de nos propres expérimentations et à l'occasion d'une compétition internationale.

Par ailleurs, nous avons utilisé notre modélisation non-fusionnée dans le cas de données de *génétique-génomique* réelles issues de la plante *Arabidopsis thaliana* afin d'en apprendre le réseau de régulation. Un premier réseau a ainsi été obtenu dont certaines régulations ont été validées d'après la littérature existante.

Table des matières

Liste de logiciels de reconstruction de réseaux vii

Introduction **1**

I Apprentissage de la structure d'un réseau bayésien 7

1 Modèles graphiques probabilistes et réseaux bayésiens **9**
1.1 Notions de probabilité 10
1.1.1 Probabilités conditionnelles 12
1.1.2 Indépendances 12
1.2 Notions sur les graphes 13
1.2.1 Graphe non-orienté 13
1.2.2 Graphe orienté 14
1.3 Réseaux bayésiens 16
1.3.1 Définition d'un réseau bayésien 16
1.3.2 Indépendances et structure du graphe 17
1.3.2.1 Condition de Markov 17
1.3.2.2 Séparations dans le graphe 18
1.3.3 Équivalent de Markov 20
1.3.4 Causalité dans les réseaux bayésiens 21
1.3.5 Énumération des réseaux bayésiens 22
1.3.6 Apprentissage des réseaux bayésiens 22
1.3.6.1 Apprentissage de la structure 22
1.3.6.2 Apprentissage des paramètres 23

2 État de l'art **25**
2.1 Apprentissage de la structure d'un réseau bayésien 26
2.1.1 Recherche d'indépendances conditionnelles 27
2.1.1.1 Mesures de corrélation 27
2.1.1.2 Algorithmes de recherche d'indépendances 29
2.1.2 Optimisation d'un score 30
2.1.2.1 Fonctions de score 30

2.1.2.2 Méthodes optimales sous contraintes . . . 33
2.1.2.3 Espace de recherche des DAG . . . 34
2.1.2.4 Espace de recherche des cpDAG . . . 39
2.1.2.5 Espace de recherche des ordres . . . 41
2.1.3 Approches hybrides . . . 42
2.2 Discrétisation des données . . . 46
2.2.1 Méthodes de discrétisations . . . 46
2.2.1.1 Méthodes univariées . . . 46
2.2.1.2 Méthodes multivariées . . . 48
2.2.2 Choix du nombre de classes . . . 49
2.3 Conclusion . . . 50

3 Nouvelle recherche stochastique gloutonne 51
3.1 Stochastic Greedy Search . . . 52
3.2 Opérateur SWAP . . . 55
3.3 Opérateurs itératifs . . . 58
3.4 Implémentation sous COMET . . . 62
3.4.1 Utilisation de COMET pour l'apprentissage des réseaux bayésiens . . . 63
3.4.2 Sources d'optimisation . . . 65
3.4.3 Inconvénients du langage COMET . . . 65
3.4.4 Analyse critique de notre implémentation . . . 66
3.5 Évaluation expérimentale des opérateurs . . . 68
3.5.1 Méthodes comparées . . . 68
3.5.2 Critères d'évaluation . . . 69
3.5.3 Réseaux et données utilisés . . . 70
3.5.4 Étude expérimentale . . . 71
3.5.4.1 Opérateurs SWAP et itératifs . . . 71
3.5.4.2 Comparaison avec d'autres approches . . . 75
3.6 Conclusion . . . 83

4 Propositions d'évolution à chaque étape de l'apprentissage 85
4.1 Nouvelles politiques de discrétisations . . . 86
4.1.1 Discrétisations univariées . . . 86
4.1.2 Discrétisation multivariée itérative . . . 87
4.2 Filtre adapté au score . . . 91
4.3 Proposition de scores étendus . . . 92
4.4 Conclusion . . . 94

Conclusion et perspectives à l'apprentissage de structure 95

II Application à la génétique-génomique 97

5 Introduction à la génétique-génomique 101
5.1 Le génome . . . 101
5.1.1 De la cellule au gène . . . 102
5.1.2 Du gène à la protéine . . . 102
5.1.3 L'expression d'un gène . . . 104
5.2 Les régulations géniques . . . 107
5.2.1 Les différents types de régulation . . . 107
5.2.2 Les différents types de réseaux de régulation . . . 108
5.2.3 Les réseaux de régulation de gènes . . . 110
5.3 Polymorphismes et variations d'expression . . . 111
5.4 Conclusion . . . 112

6 Apprentissage d'un réseau de régulation de gènes 115
6.1 Caractéristiques des données . . . 116
6.1.1 Observations statiques versus temporelles . . . 116
6.1.2 Données discrètes versus continues . . . 117
6.1.3 Variabilité des données d'expression . . . 118
6.2 Approches existantes à partir de données d'expression . . . 120
6.2.1 Méthodes non-paramétriques . . . 121
6.2.1.1 Dépendances statistiques . . . 121
6.2.1.2 Arbres de régression . . . 124
6.2.2 Modèles graphiques gaussiens . . . 125
6.2.3 Réseaux bayésiens . . . 130
6.3 Approches intégratives . . . 131
6.4 Conclusion . . . 134

7 Comparaisons des approches 135
7.1 Modélisations proposées pour les réseaux bayésiens . . . 136
7.1.1 Modèle non-fusionné . . . 137
7.1.2 Modèle fusionné . . . 140
7.2 Comparaison des approches . . . 142
7.2.1 Comparaisons sur les réseaux Web-50 . . . 144
7.2.1.1 Réseaux de régulations utilisés . . . 144
7.2.1.2 Génération des données de génétique-génomique . . . 145
7.2.1.3 Critères d'évaluation . . . 147
7.2.1.4 Méthodes comparées . . . 147
7.2.1.5 Résultats . . . 147
7.2.2 Compétition DREAM . . . 155
7.2.2.1 Présentation de la compétition DREAM . . . 155
7.2.2.2 Données du challenge . . . 155

7.2.2.3 Critères d'évaluation . . . 157
7.2.2.4 Méthodes comparées . . . 158
7.2.2.5 Résultats officiels . . . 161
7.2.2.6 Résultats supplémentaires . . . 168
7.3 Conclusion . . . 172

8 Application aux données d'*Arabidopsis thaliana* 173
8.1 *Arabidopsis thaliana* . . . 173
8.2 Analyse classique : recherche d'eQTLs . . . 174
8.3 Méthodologie employée . . . 174
8.3.1 Complétions des données . . . 174
8.3.2 Réglage de notre approche . . . 175
8.4 Résultats . . . 176
8.4.1 Caractéristiques du réseau . . . 176
8.4.2 Nomenclature des cas . . . 178
8.4.3 Comptage des situations . . . 180
8.4.4 Comparaisons et validation à partir de bases de connaissances biologiques . . . 180
8.5 Conclusion . . . 183

Perspectives pour l'application aux données biologiques 185

Remarques générales 187

Liste de logiciels de reconstruction de réseaux

ARACNE, 123

BANJO, 143

CLR, 123

GeneNet, 128
GES, 40
GGMselect, 128

LAGD, 37

MMHC, 45

ParCorA, 126

Régressions Lasso / LARS, 126

Sélecteur de Dantzig / glpk, 160
SCT, 134
SGS / Comet, 52
SIMoNe, 128

Introduction

Apprendre la structure d'un réseau bayésien appliqué aux données de *génétique-génomique*, voilà donc l'objet de cette thèse, avant de rentrer dans le vif du sujet arrêtons nous sur les différents éléments de ce titre.

Apprendre tout d'abord, voilà une chose que l'on fait depuis notre naissance, nous apprenons tout le temps et sur tout. Sans nécessairement rentrer dans des concepts philosophiques du "*Qu'est ce qu'apprendre ?*", nous pouvons voir l'apprentissage comme la compréhension des éléments qui nous entoure, que ces éléments soient physiques, palpables ou bien plus immatériels. On apprend alors aussi bien à lacer ses chaussures que les règles morales et sociales de la société qui nous entoure.

Ainsi apprendre c'est comprendre, ce qui constitue bien l'un des objectifs de la recherche toutes disciplines confondues où l'on tente d'apprendre des phénomènes encore inconnus ou mal compris. Dans le cadre de l'informatique l'apprentissage vise généralement (ou du moins dans notre cas) à comprendre un phénomène physique (le système). Cependant ce système n'est pas le résultat d'un simple programme. Il convient donc dans un premier temps, de définir un modèle mathématique permettant de représenter au plus près le phénomène. Puis de développer des méthodes (des algorithmes) afin d'apprendre les différents paramètres de ce modèle d'après un ensemble d'observations du phénomène.

Avant de définir le phénomène qui nous intéresse, présentons tout d'abord le modèle utilisé.

Apprendre oui, mais quoi ?

Parmi les modèles mathématiques employés afin de représenter des systèmes physiques nous utilisons celui des réseaux bayésiens. Ces réseaux appartiennent à la famille des modèles graphiques probabilistes. Comme son nom le laisse supposer les modèles de cette famille sont constitués d'une structure, le *graphe*, composé d'un ensemble de variables reliées entre elles par des arêtes (orientées ou non) et d'un ensemble de probabilités caractérisant les interactions entre les variables reliées dans ce graphe. En plus de ces premières caractéristiques, les réseaux bayésiens imposent une orientation des arêtes du graphe (appelées alors arcs) et l'absence de cycle orienté dans celui-ci. De plus nous considérons uniquement le cas des réseaux bayésiens discrets et statiques, c'est-à-dire que l'état de chaque variable prend ses valeurs dans un ensemble de taille fini et que nous ne prenons pas en compte le caractère temporel des données.

L'orientation du graphe permet de modéliser des systèmes présentant des relations de cause à effet, l'état d'une variable du graphe dépend alors de l'effet des variables pointant vers elle au travers d'un

arc. L'absence de cycle simplifie quant à elle le modèle, en effet on imagine la difficulté supplémentaire d'interprétation lorsque une variable X est à la fois la cause d'une variable Y tout en en subissant ses effets. En outre l'utilisation de modèles probabilistes permet de représenter le caractère incertain de ces effets dans le graphe, elle offre de fait une souplesse et une interprétation naturelle du réseau. Cette propriété est d'autant plus intéressante lorsque le phénomène n'est que partiellement observé, les probabilités nous offrent ainsi un moyen efficace de représenter l'incertitude liée à des causes non observées.

Les réseaux bayésiens permettent ainsi de représenter des systèmes de manière visuelle sous la forme d'un graphe et interprétable au travers des probabilités. Leur apprentissage consiste à déterminer leur structure (l'ensemble des arcs) et les probabilités associées, à partir d'un ensemble d'observations du phénomène d'intérêt. Dans cette thèse nous supposons détenir des observations complètes sans erreur, ainsi sous cette hypothèse, nous voyons l'apprentissage des probabilités comme secondaire et concentrons nos efforts sur l'apprentissage du graphe. Ce dernier, indépendamment de la nature du phénomène, représente une tâche difficile, la première partie de cette thèse y sera consacrée. Nous verrons à cette occasion les différentes approches proposées afin de résoudre ce problème. L'une d'entre elles consiste à explorer l'espace des graphes possibles afin de sélectionner celui qui maximise un critère, appelé *score*. Ce score mesure la capacité d'un réseau à expliquer les observations du phénomène d'intérêt, autrement dit sa *vraisemblance*, moyennant un compromis sur la complexité du modèle. Cependant la taille de l'espace des graphes possibles augmente rapidement avec le nombre de variables, il convient alors de définir des stratégies permettant de se déplacer intelligemment dans celui-ci. C'est dans cette optique que nous proposons de nouveaux opérateurs permettant de se déplacer plus efficacement dans ce large espace.

L'apprentissage de la structure d'un réseau bayésien représente ainsi un challenge en soi, cependant apprendre un réseau générique n'a d'intérêt qu'à la vue des applications possibles de ce dernier. Nous définissons donc la raison motivant notre apprentissage.

Apprendre oui, mais pourquoi ?

Dans cette thèse nous nous sommes intéressés au phénomène de régulation entre gènes.

L'activité cellulaire au sein d'un organisme vivant est complexe et comporte de nombreux acteurs tels que les *gènes*, les *protéines*, les *ARNs* ou les *métabolites* qui interagissent à différents niveaux de la cellule. Les régulations entre gènes représentent l'un de ces niveaux. Les gènes placés le long des chromosomes produisent pour la majorité d'entre eux des molécules appelées les ARNs, une partie de ces ARNs étant ensuite traduite en protéines. Cette quantité de protéine ou d'ARN produite par le gène caractérise son *niveau d'expression*. Les ARNs et les protéines remplissent de nombreux rôles dans la cellule et leur niveau anormalement bas ou élevé dans la cellule peut être le signe de stress ou de maladie pour l'organisme. Parmi ces rôles nous nous intéressons à celui de régulateur de l'expression des gènes. En effet les protéines ou ARNs produits par un gène viennent réguler le niveau d'expression d'autres gènes définissant ainsi le phénomène de *régulations*. Ces régulations

pour l'ensemble des gènes forment alors le *réseau de régulation de gènes* de la cellule que l'on peut représenter sous la forme d'un graphe orienté où à chaque nœud correspond un gène et à chaque arc, une régulation. Connaître ce réseau permet de mieux comprendre l'activité de la cellule et de prévoir au mieux les conséquences dues à une perturbation naturelle ou artificielle d'un gène. Cependant construire l'intégralité du réseau de régulation à partir des niveaux d'expression des gènes est un challenge difficile. De nombreuses approches ont été proposées dans ce but, dont certaines utilisent notamment les réseaux bayésiens, cependant ces méthodes n'étudient généralement qu'un sous ensemble de gènes et bien peu des régulations prédites sont validées expérimentalement. Ainsi malgré ces nombreux travaux et avancées la tâche reste difficile dû à la fois à la taille de ces réseaux, composés de plusieurs milliers de gènes et à la complexité du phénomène de régulation.

En effet l'intensité de ces régulations varie en fonction de plusieurs facteurs. Une reconstruction efficace des régulations passe donc par la prise en compte de ces facteurs. L'un d'entre eux provient de la variabilité génétique des individus. Cette variabilité s'exprime par des mutations au niveau de la séquence d'ADN, lorsqu'une mutation se situe au niveau d'un gène X, deux types d'effet peuvent alors apparaître. La première situation correspond à un effet direct de la mutation sur l'expression du gène X. Dans l'autre situation la mutation vient modifier la structure de la protéine générée par le gène X, sans pour autant changer son niveau d'expression. Ainsi l'effet de cette dernière mutation sera déporté sur les gènes régulés par le gène X. Nous percevons ainsi la complexité du phénomène de régulation et l'intérêt d'utiliser cette information sur la présence de mutations afin d'améliorer la reconstruction du réseau de régulation.

Ces données contenant à la fois les niveaux d'expression et la présence de mutations sur la séquence d'ADN sont appelées données de *génétique-génomique*. Dans cette thèse nous nous attachons donc à présenter une modélisation du problème d'apprentissage de réseau de régulation de gènes pour ce type de données dans le cadre des réseaux bayésiens. Ainsi apprendre la structure d'un réseau de régulation de gènes revient à apprendre le réseau bayésien correspondant.

Par ailleurs les postulats émis concernant les données de *génétique-génomique* sont plus réalistes dans le cas de populations contrôlées. Les plantes représentent des organismes de référence pour ce type de population, nous verrons donc une application de notre approche à des données réelles provenant de la plante *Arabidopsis thaliana*.

Organisation du document

Cette thèse se décompose donc en deux parties. La première partie présente le problème d'apprentissage de la structure de réseaux bayésiens discrets génériques. La seconde décrit l'application de cet apprentissage au cas des réseaux de régulation de gènes.

Nous débutons cette première partie par un chapitre dédié aux notions essentielles de probabilité et de la théorie des graphes afin de mieux comprendre le concept des réseaux bayésiens. Nous décrivons à cette occasion quelques propriétés de ces réseaux et notamment le lien existant entre l'indépendance

des variables et la structure du graphe.

Le chapitre 2 présente un état de l'art sur les méthodes d'apprentissage de structure de réseaux bayésiens discrets. Ces méthodes peuvent être classées suivant la stratégie employée, nous distinguons ainsi les approches recherchant directement les indépendances portées par les observations de celles utilisant un score afin de guider une exploration dans l'espace des graphes. Nous décrivons également une troisième stratégie dite *hybride* qui consiste à utiliser de concert les deux approches précédentes. Nous abordons finalement le problème de la discrétisation des données, fréquent lorsque les observations sont issues de phénomènes physiques et nécessaire à l'utilisation des réseaux bayésiens discrets.

Le chapitre 3 expose nos premières contributions dans le cadre des approches utilisant un score, en débutant par une extension stochastique de l'algorithme Greedy Search que nous appelons SGS. Nous présentons également deux nouveaux opérateurs afin d'éviter certaines situations d'optima locaux du score. Le premier, appelé SWAP, permet de changer pour une variable l'un de ses parents en une seule opération, tandis que le second consiste en une extension itérative des opérateurs existants, qui autorise temporairement l'exploration des graphes cycliques. Nous détaillons par la suite l'implémentation de SGS et des deux opérateurs dans le langage COMET avant de présenter une série de comparaisons face à plusieurs méthodes de l'état de l'art.

Dans le chapitre 4 nous décrivons une seconde série de contributions à l'apprentissage de structure portant sur des points plus ciblés. Nous débutons par deux méthodes de discrétisation, adaptée pour la première au cas de données biologiques, tandis que la seconde plus générique, établit un ensemble de discrétisations minimisant la perte d'information mutuelle entre les variables. Puis nous proposons dans un second temps un filtre permettant de réduire l'espace de recherche afin d'apprendre des graphes comptant plusieurs milliers de variables. Enfin nous définissons l'idée de scores étendus afin d'établir un *a priori* uniforme sur les classes de connectivité plutôt que sur l'ensemble des graphes. Ce chapitre clôture par ailleurs la première partie du manuscrit.

La seconde partie est dédiée à l'application biologique de cette thèse, à savoir l'apprentissage de la structure d'un réseau de régulation de gène.

Dans le chapitre 5 nous introduisons les notions de base de la biologie cellulaire afin de préciser le phénomène de régulation et les différents niveaux auquel ce dernier agit. Nous présentons à cette occasion les données dites de *génétique-génomique* et l'intérêt de leur utilisation pour la reconstruction de réseaux de régulation.

Le chapitre 6 se focalise justement sur les méthodes développées afin d'apprendre la structure des réseaux de régulation. Après avoir précisé les caractéristiques des données d'expression de gènes, nous présentons les approches existantes utilisant ces données d'expression seules. Puis nous décrivons celles qui utilisent les données de *génétique-génomique*.

Le chapitre 7 constitue la 3$^{\text{ème}}$ série de nos contributions. Nous proposons tout d'abord deux modélisations possibles du problème de reconstruction de réseaux de régulation pour les données de *génétique-génomique* dans le cadre des réseaux bayésiens. A savoir un modèle non-fusionné représentant les deux informations de manière séparée tandis que le second modèle propose de fusionner ces deux informations. Puis nous comparons ces deux modèles dans des situations variées. Dans un premier temps nous utilisons des réseaux de taille modeste en confrontant nos modèles à des

méthodes de l'état de l'art. Dans un second temps nous comparons un seul des deux modèles face à trois autres approches développées par l'équipe SaAB dans le cadre d'une compétition internationale sur des réseaux comportant 2 000 variables.

Le chapitre 8 clôture cette deuxième partie en présentant l'application de l'un de nos modèles à des données réelles de *génétique-génomique* provenant de la plante *Arabidopsis thaliana*. Nous commençons par décrire les différents pré-traitements et modifications nécessaires de notre approche pour ces données. Nous analysons par la suite le réseau obtenu et nous comparons celui-ci à une approche plus classique basée sur la recherche d'eQTLs. Finalement nous validons quelques configurations apprises par notre approche grâce à la littérature.

Première partie

Apprentissage de la structure d'un réseau bayésien

Chapitre 1

Modèles graphiques probabilistes et réseaux bayésiens

Sommaire

1.1 Notions de probabilité 10
1.1.1 Probabilités conditionnelles 12
1.1.2 Indépendances 12
1.2 Notions sur les graphes 13
1.2.1 Graphe non-orienté 13
1.2.2 Graphe orienté 14
1.3 Réseaux bayésiens 16
1.3.1 Définition d'un réseau bayésien 16
1.3.2 Indépendances et structure du graphe 17
1.3.2.1 Condition de Markov 17
1.3.2.2 Séparations dans le graphe 18
1.3.3 Équivalent de Markov 20
1.3.4 Causalité dans les réseaux bayésiens 21
1.3.5 Énumération des réseaux bayésiens 22
1.3.6 Apprentissage des réseaux bayésiens 22
1.3.6.1 Apprentissage de la structure 22
1.3.6.2 Apprentissage des paramètres 23

Ce chapitre décrit les notions essentielles à la compréhension des modèles graphiques probabilistes et plus précisément des réseaux bayésiens. Nous introduisons notamment les concepts de base des probabilités et de la théorie des graphes. Ce chapitre permet de familiariser le lecteur aux réseaux bayésiens mais ne vise en aucun cas à présenter de manière exhaustive les différentes facettes de ces réseaux, dans ce but nous pouvons conseiller la lecture de Naim et al. [2007], Koller and Friedman [2009].

1.1 Notions de probabilité

Les probabilités sont utilisées continuellement, de manière plus ou moins consciente, afin d'exprimer notre croyance sur le fait qu'un évènement se produise. L'idée d'événement est un concept générique pouvant prendre de multiples formes. On décrit de la même manière la probabilité de tirer un roi de cœur dans un jeu de carte que celle portant sur la chute de neige au mois de décembre 2012. Ces probabilités sont toutes définies dans l'intervalle [0,1], quantifiant ainsi notre niveau de croyance en l'évènement.

Plus formellement supposons un évènement ω appartenant à l'ensemble des évènements observables possibles Ω. On peut alors définir une fonction de probabilité

$$\mathbb{P} : \omega \to [0, 1] \quad \omega \in \Omega$$

où
- $\mathbb{P}(\omega) \geq 0$ représente la probabilité que ω se produise ;
- $\mathbb{P}(\Omega) = 1$ dénote le caractère certain qu'au moins un des évènements possibles se produise ;
- $\mathbb{P}(\omega_1 \cup \omega_2) = \mathbb{P}(\omega_1) + \mathbb{P}(\omega_2)$ spécifie l'additivité des probabilités de deux évènements disjoints ω_1, ω_2.

Interprétation des probabilités La valeur de ces probabilités peut être interprétée de deux manières. La première correspond à une vision fréquentielle des probabilités. On définit alors la probabilité d'évènements issus d'expériences pouvant être répétées. Ainsi la probabilité de tirer un roi de cœur dans un jeu de carte classique, correspond à la fréquence d'apparition de cette carte si on répète indéfiniment un tirage avec remise dans le jeu. Cette interprétation permet de définir avec précision la valeur des probabilités, cependant dans ce cas, seules des probabilités touchant à des systèmes physiques et concrets peuvent être définies. La seconde vision est donc plus générale et utilise la notion de croyance subjective de l'utilisateur afin de déterminer ces probabilités. Prenons l'exemple de la probabilité de chutes de neige au mois de décembre, la valeur de cette probabilité ne peut être déterminée par une approche fréquentielle du fait que cet évènement est unique et ne peut être répété. Les prévisions météorologiques tentent cependant d'estimer ce type de probabilité de manière fréquentielle en considérant par exemple les précédentes situations climatiques similaires comme des répétitions d'une même expérience. Malgré tout, la définition des règles, permettant de mesurer la similarité entre les situations antérieures, est laissée à l'interprétation des prévisionnistes et à leur connaissance des phénomènes météorologiques.

L'interprétation de ces probabilités dépend donc du contexte des évènements observés, par la suite nous nous intéresserons uniquement aux valeurs de ces probabilités sans s'attacher à leur interprétation.

Variables aléatoires La notion d'évènement bien que naturelle amène parfois à définir un ensemble d'évènements possibles (noté Ω) extrêmement large. Il est toutefois possible de décomposer ces évènements en définissant des attributs prenant la forme de variables aléatoires. Reprenons l'exemple

du tirage d'un roi de cœur dans un paquet standard de 52 cartes à jouer. L'ensemble Ω comporte ici 52 évènements possibles correspondant au tirage de l'une des 52 cartes distinctes du paquet. L'évènement "tirer un roi de cœur" peut se décomposer en deux évènements distincts qui sont "tirer un roi" et "tirer un cœur". On définit ainsi les attributs "figure" et "enseigne" afin de caractériser la carte tirée. Chacun de ces attributs peut être représenté sous la forme d'une variable aléatoire X correspondant à une fonction définie telle que

$$X : \omega \to X(\omega) \quad X(\omega) \in D_X$$

où D_X représente le domaine de définition de X.

Cette fonction associe donc à chaque évènement ω, la valeur $x = X(\omega)$ de l'attribut représenté par X. Dans notre exemple nous avons ainsi deux variables aléatoires *Figure* et *Enseigne* de domaines $D_{Figure} = \{As, Roi, Dame, Valet, 10, 9, 8, 7, 6, 5, 4, 3, 2\}$ et $D_{Enseigne} = \{Carreau, Coeur, Pique, Trèfle\}$.

On note $\mathbb{P}(X = x)$ la probabilité qu'une variable aléatoire X prenne la valeur x. En l'absence d'ambiguïté nous l'écrirons de manière concise $\mathbb{P}(x)$, on a ainsi

$$\sum_{x \in D_X} \mathbb{P}(x) = 1$$

La distribution de probabilités de la variable X est dite multinomiale et associe à chaque valeur du domaine de X sa probabilité de réalisation ($x \to \mathbb{P}(x)$).

Probabilités jointes et marginales Calculer la probabilité de l'évènement "tirer un roi de cœur" à partir des deux variables aléatoires *Figure* et *Enseigne* revient alors à calculer la probabilité *jointe* de tirer une carte qui soit à la fois un roi et un cœur.

La probabilité *jointe* de deux variables aléatoires X et Y est définie sur $D_X \times D_Y$ par la fonction $\mathbb{P}$ telle que

$$\mathbb{P} : (x, y) \to \mathbb{P}(x, y)$$

où $\mathbb{P}(x, y) = \mathbb{P}(X = x \cap Y = y)$

Dans notre exemple la probabilité *jointe* de l'évènement "tirer un roi de cœur" s'écrit alors $\mathbb{P}(Roi, Coeur)$. Ces probabilités peuvent être étendues à un ensemble fini de variables aléatoires.

A l'inverse, il est possible de réduire l'ensemble des variables aléatoires d'une probabilité jointe en la *marginalisant*. Considérons une probabilité jointe $\mathbb{P}$ sur un ensemble U de variables aléatoires. Soit V et V' deux sous-ensembles de U tels que $V' = U \setminus V$, on définit alors la *marginalisation* de $\mathbb{P}$ sur V par

$$\mathbb{P}(v) = \sum_{v' \in D_{V'}} \mathbb{P}(v, v') \quad v \in D_V$$

On peut ainsi dans notre exemple *marginaliser* $\mathbb{P}(Roi, Coeur)$ sur la variable *Figure* afin d'obtenir $\mathbb{P}(Roi) = \mathbb{P}(Roi, Carreau) + \mathbb{P}(Roi, Coeur) + \mathbb{P}(Roi, Pique) + \mathbb{P}(Roi, Trèfle)$ soit $\mathbb{P}(Roi) = \frac{4}{52} = \frac{1}{13}$.

1.1.1 Probabilités conditionnelles

Supposons maintenant que nous tirons notre carte uniquement dans l'ensemble des cartes de cœur. La probabilité de tirer un roi de cœur n'est donc plus $\mathbb{P}(Roi, Coeur)$ mais uniquement la probabilité de tirer un roi sachant que la carte sera un cœur. Cette probabilité dite *conditionnelle* s'écrit alors $\mathbb{P}(Roi|Coeur)$.

Plus formellement pour deux variables aléatoires X et Y, la probabilité *conditionnelle* de $X = x$ sachant $Y = y$ s'écrit

$$\mathbb{P}(x|y) = \frac{\mathbb{P}(x, y)}{\mathbb{P}(y)}$$

Y est alors appelée variable de *conditionnement*, cette notion se généralise à un ensemble fini de variables de *conditionnement* et par convention $\mathbb{P}(x|\emptyset) = \mathbb{P}(x)$.

Dans notre exemple $\mathbb{P}(Roi|Coeur) = \frac{\mathbb{P}(Roi,Coeur)}{\mathbb{P}(Coeur)} = \frac{\frac{1}{52}}{\frac{13}{52}} = \frac{1}{13}$. On remarque ici que $\mathbb{P}(Roi|Coeur) = \mathbb{P}(Roi) = \frac{1}{13}$, on en déduit dans ce cas que les variables *Figure* et *Enseigne* sont indépendantes comme nous le préciserons dans la section 1.1.2.

Loi fondamentale généralisée La probabilité jointe sur un ensemble de p variables $X_{i\in[1..p]}$ peut s'écrire à l'aide de ces probabilités conditionnelles sous la forme

$$\mathbb{P}(x_1, x_2, .., x_p) = \prod_{i=1}^{p} \mathbb{P}(x_i|x_1..x_{i-1})$$

Théorème de Bayes De même on obtient pour deux variables aléatoires X et Y, l'égalité

$$\mathbb{P}(x, y) = \mathbb{P}(x|y)\mathbb{P}(y) = \mathbb{P}(y|x)\mathbb{P}(x)$$

généralement écrite sous la forme du théorème de Bayes

$$\mathbb{P}(x|y) = \frac{\mathbb{P}(y|x)\mathbb{P}(x)}{\mathbb{P}(y)}$$

Cette écriture permet ainsi de passer d'une probabilité conditionnelle à sa probabilité conditionnelle inverse.

1.1.2 Indépendances

Nous avons vu précédemment que $\mathbb{P}(Roi|Coeur) = \mathbb{P}(Roi)$, ainsi la connaissance de l'enseigne de la carte ne change pas la probabilité de tirer un roi. Cette observation reste valide pour toutes les figures et toutes les enseignes possibles. Les deux variables *Enseigne* et *Figure* sont alors dites *indépendantes*.

On note que deux variables aléatoires X et Y sont *indépendantes* (noté $X \perp Y$) si et seulement si

$$\mathbb{P}(x|y) = \mathbb{P}(x) \quad \forall x \in D_X, \forall y \in D_Y$$

Cette notion d'indépendance s'étend au cas des probabilités conditionnelles. Ainsi deux variables aléatoires X et Y sont indépendantes sachant une troisième variable Z (noté $X \perp Y|Z$) si et seulement si

$$\mathbb{P}(x|y,z) = \mathbb{P}(x|z) \quad \forall x \in D_X, \forall y \in D_Y, \forall z \in D_Z$$

On peut également généraliser l'indépendance conditionnelle au cas d'ensembles de variables. Ainsi deux ensembles $\mathcal{X}$ et $\mathcal{Y}$ de variables aléatoires sont indépendants sachant un troisième ensemble $\mathcal{Z}$ (noté $\mathcal{X} \perp \mathcal{Y}|\mathcal{Z}$) si et seulement si

$$X \perp Y|\mathcal{Z} \quad \forall X \in \mathcal{X}, \forall Y \in \mathcal{Y}$$

L'indépendance conditionnelle permet de réduire le calcul de la loi fondamentale généralisée. Chaque probabilité conditionnelle se simplifie alors tel que

$$\mathbb{P}(x_i|x_1..x_{i-1}) = \mathbb{P}(x_i|U_i) \quad i = [1..p]$$

où $U_i \subseteq \{X_1, .., X_{i-1}\}$ tel que $X_i \perp \{X_1, .., X_{i-1}\} \setminus U_i | U_i$.

U_i représente donc un sous-ensemble des variables précédant X_i, qui permet de rendre X_i indépendante des variables la précédant (hormis celles inclues dans U_i).

1.2 Notions sur les graphes

Après avoir introduit les concepts de base sur les probabilités, nous décrivons ici quelques éléments caractéristiques des graphes en distinguant le cas des graphes non-orientés de celui des graphes orientés.

1.2.1 Graphe non-orienté

Un graphe non-orienté $\mathcal{G}$ (ou non-dirigé) est constitué d'un ensemble de nœuds $\mathcal{X}$ reliés entre eux par un ensemble d'arêtes $\mathcal{E}$ tel que deux nœuds peuvent être reliés par au plus une arête. Deux nœuds reliés par une arête dans $\mathcal{G}$ sont dit *adjacents*, on note ainsi $Adj_{\mathcal{G}}(X_1)$ l'ensemble des nœuds adjacents à X_1 dans $\mathcal{G}$. Cet ensemble constitue le *voisinage* de X_1.

La Figure **1.1** représente un graphe $\mathcal{G}$ non-orienté composé de 7 nœuds, où $Adj_{\mathcal{G}}(X_1) = \{X_2\}$ ou encore $Adj_{\mathcal{G}}(X_7) = \{X_5, X_6\}$. Nous utilisons ce graphe afin d'illustrer la terminologie décrite par la suite.

Chaînes et cycles On appel *chaîne* toute succession d'arêtes distinctes dans $\mathcal{G}$ reliant un ensemble de nœuds $\{X_1, ..., X_k\}$ tel que pour tout $i = \{1, ..., k-1\}$ on a l'arête $X_i - X_{i+1}$. Un *cycle* est alors une chaîne de taille supérieure à 2 où $X_1 = X_k$. Dans le cas particulier d'un cycle de taille 1 tel que $X_1 - X_1$, on utilise le terme de *boucle*.

Les ensembles $\{X_2 - X_4, X_4 - X_3\}$ et $\{X_5 - X_6, X_6 - X_7, X_7 - X_5\}$ constituent ainsi des chaînes de taille respective 2 et 3 où seule la seconde forme un cycle.

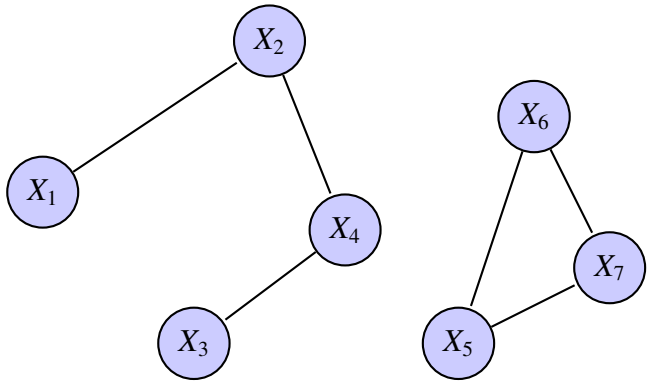


Figure 1.1 – Exemple d'un graphe non-orienté.

Graphes connexes et graphes complets Un graphe $\mathcal{G}$ est dit *connexe* si chaque nœud de $\mathcal{G}$ peut être relié par une chaîne à tout autre nœud de $\mathcal{G}$. De manière plus forte $\mathcal{G}$ est dit *complet* si chaque nœud de $\mathcal{G}$ est adjacent à tout autre nœud de $\mathcal{G}$, dans ce cas $\mathcal{G}$ forme une *clique*.

Ainsi le graphe de la Figure **1.1** n'est pas connexe puisqu'il est par exemple impossible de relier X_4 à X_5. En considérant les deux sous-graphes $\mathcal{G}_1$ et $\mathcal{G}_2$ constitués respectivement des ensembles de nœuds $\{X_1, X_2, X_3, X_4\}$ et $\{X_5, X_6, X_7\}$, on obtient alors deux graphes connexes, de plus $\mathcal{G}_2$ est complet.

Arbres et forêts Dans le cadre des graphes non-orientés, un *arbre* est un graphe connexe sans cycle, tandis qu'une *forêt* est constituée d'un ensemble d'arbres. Dans notre exemple $\mathcal{G}$ ne représente pas une forêt dû au cycle présent dans le sous-graphe $\mathcal{G}_2$, seul $\mathcal{G}_1$ est un arbre.

1.2.2 Graphe orienté

Appliquons de manière symétrique ces définitions au cas des graphes orientés.

Un graphe orienté $\mathcal{G}$ (ou dirigé) est constitué d'un ensemble de nœuds $\mathcal{X}$ reliés entre eux par un ensemble d'*arcs* $\mathcal{E}$. Pour chaque arc $X_i \rightarrow X_j$, X_i représente le nœud *parent* et X_j le nœud *enfant* (ou *fils*). L'ensemble des parents de X_i est noté $Pa(X_i)$ tandis que $Chi(X_j)$ représente l'ensemble des enfants de X_j, le voisinage de X_i s'écrit alors comme l'union de $Pa(X_i)$ et de $Chi(X_i)$. On définit également les *ancêtres* de X_j (notés $Anc(X_j)$) comme l'ensemble des nœuds construit de manière itérative à partir de $Pa(X_j)$, en incluant à chaque itération les parents de chaque ancêtre. De manière symétrique on définit la *descendance* de X_i (notée $Desc(X_i)$) en recherchant à partir de $Chi(X_i)$ les enfants de chaque descendant.

Suivant la Figure **1.2**, on a notamment $Pa(X_7) = \{X_5, X_6\}$ ou encore $Desc(X_2) = \{X_1, X_3, X_4\}$.

Chemins et circuits On appel *chemin* toute succession d'arcs distincts dans $\mathcal{G}$ reliant un ensemble de nœuds $\{X_1, ..., X_k\}$ tel que pour tout $i = \{1, ..., k-1\}$ on a $X_i \rightarrow X_{i+1}$. Ainsi un *circuit* est un chemin où $X_1 = X_k$. On remarque ici, à la différence des graphes non-orientés, qu'un circuit peut être composé des deux seuls arcs $X_1 \rightarrow X_k$ et $X_k \rightarrow X_1$.

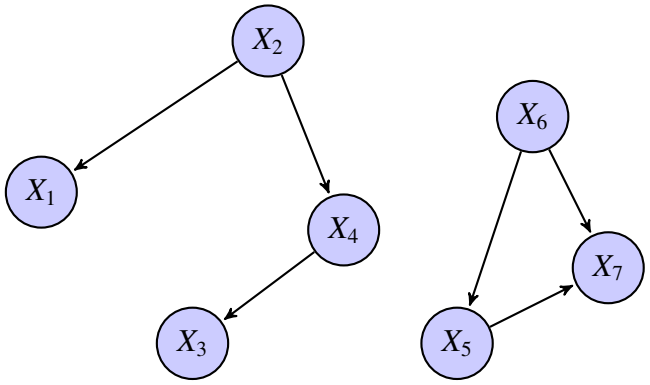


Figure 1.2 – Exemple de graphe orienté.

L'ensemble $\{X_6 \rightarrow X_5, X_5 \rightarrow X_7\}$ constitue ainsi un chemin de taille 2, mais aucun circuit n'est présent dans le graphe.

Nous noterons par ailleurs que le terme *cycle* peut également être utilisé pour les graphes orientés, son sens dépend alors du type de graphe considéré (orienté ou non). Nous maintiendrons dans cette thèse le terme *chemin* afin de lever toute ambiguïté, cependant nous utiliserons les dérivés du terme *cycle* à savoir *acyclicité* et *acyclique* afin de caractériser l'absence de circuit dans un graphe orienté.

Graphes connexes et graphes complets Les notions de connexité et de complétude ne s'appliquent pas aux graphes orientés en remplaçant directement le concept de chaîne par celui de chemin, il est ainsi nécessaire de revenir à l'état d'un graphe non-orienté. Pour cela on définit le *squelette* d'un graphe orienté $\mathcal{G}$ comme le graphe non-orienté obtenu en convertissant chaque arc $X_i \rightarrow X_j$ en une arête $X_i - X_j$. Un graphe orienté est donc connexe ou complet si son squelette l'est.

De plus dans le cas des graphes orientés, un graphe est dit *fortement connexe* s'il existe pour tout couple (X_i,X_j) à la fois un chemin allant de X_i vers X_j mais aussi un chemin de X_j vers X_i.

Arborescence et forêts Une arborescence est un arbre orienté à partir d'une *racine*. Ainsi chaque arête $X_i - X_j$ de l'arbre est orientée en $X_i \rightarrow X_j$ si la chaîne reliant la racine à X_i ne passe pas par X_j et $X_i \leftarrow X_j$ sinon. Une arborescence est alors un graphe orienté tel qu'il existe un chemin unique de la racine vers n'importe quel autre nœud du graphe. Une *forêt* représente un ensemble d'arborescences. Dans notre exemple, seul $\mathcal{G}_1$ est une arborescence de racine X_2.

Notons finalement que certains graphes sont composés à la fois d'arcs et d'arêtes, on parle alors de graphes *partiellement orientés* ou graphes *mixtes*.

1.3 Réseaux bayésiens

1.3.1 Définition d'un réseau bayésien

Les réseaux bayésiens font parti de la famille des modèles graphiques probabilistes au même titre que les champs de Markov [Besag, 1974]. Ces réseaux permettent de représenter de manière concise la distribution de probabilité jointe sur un ensemble de variables aléatoires. Il est possible de rencontrer dans la littérature différentes dénominations pour ces réseaux, telles que *les réseaux probabilistes* ou *les réseaux de croyance*.

Ce concept introduit par Pearl [1985], permet de définir un réseau bayésien $\mathcal{B}$ comme un couple $\{\mathcal{G}, \mathcal{P}\}$ où $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ est un graphe dirigé sans circuit (ou DAG pour *Directed Acyclic Graph*) et $\mathcal{P}$ une collection de distributions de probabilités conditionnelles pour chaque variable aléatoire de $\mathcal{X}$ tel que la distribution de la probabilité jointe s'écrit

$$\mathbb{P}(X_1, .., X_p) = \prod_{i=1}^{p} \mathbb{P}(X_i | Pa(X_i))$$

où $Pa(X_i)$ est l'ensemble des parents de X_i dans $\mathcal{G}$.

On remarque la similarité de cette expression avec la loi fondamentale généralisée vue précédemment et plus particulièrement après sa simplification à l'aide des ensembles de conditionnement restreints U_i. En supposant un ordre sur les variables aléatoires tel que $Pa(X_i) \subseteq \{X_1, .., X_{i-1}\}$, on obtient exactement la même expression pour $U_i = Pa(X_i)$. On dira dans ce cas que $\mathbb{P}$ se factorise suivant $\mathcal{G}$ et que $\mathcal{G}$ représente $\mathbb{P}$. On perçoit ainsi la relation forte entre la structure de $\mathcal{G}$ et les indépendances conditionnelles présentes dans $\mathbb{P}$.

On note $Dim(\mathcal{B})$ la dimension du réseau bayésien $\mathcal{B}$ tel que $Dim(\mathcal{B}) = \sum_{i=1}^{p} Dim(B_{X_i})$ avec $Dim(B_{X_i}) = (r_i - 1)q_i$ où r_i est égal à la taille du domaine D_{X_i} de la variable X_i et $q_i = \prod_{X_j \in Pa(X_i)} r_j$ correspond au nombre de configurations possibles pour les parents de X_i.

Différentes extensions des réseaux bayésiens ont été proposées pour des applications spécifiques, on peut ainsi citer les réseaux bayésiens orientés objet [Bangsø and Wuillemin, 2000] ou encore les diagrammes d'influence [Jensen and Nielsen, 2007]. L'extension la plus notable est probablement celle des réseaux bayésiens *dynamiques*, proposée par Dean and Kanazawa [1989], qui se distinguent des réseaux bayésiens classiques dits *statiques*.

Réseaux bayésiens dynamiques Un réseau bayésien *dynamique* permet de prendre en compte l'aspect temporel des variations de ses variables. Ce type de réseau s'organise en couche comme nous pouvons le voir sur la Figure **1.3**.

Chaque couche permet de représenter l'état des variables à un instant donné t, qui dépend uniquement de l'état des variables aux instants antérieurs. Dans notre exemple le réseau respecte

notamment la *propriété de Markov de 1^er ordre* du fait que l'état à l'instant t ne dépend que de l'instant $t-1$ c'est-à-dire que $\mathbb{P}(X_i^t|\mathcal{X}^1,..,\mathcal{X}^{t-1}) = \mathbb{P}(X_i^t|\mathcal{X}^{t-1})$ avec X_i^t l'état de la variable X_i à l'instant t et $\mathcal{X}^t = \{X_1^t,..,X_p^t\}$. De même on parle de réseau *homogène* lorsque les transitions entre variables restent constantes au cours du temps d'où $\mathbb{P}(\mathcal{X}^{t+2}|\mathcal{X}^{t+1}) = \mathbb{P}(\mathcal{X}^{t+1}|\mathcal{X}^t)$. Ces propriétés peuvent être relâchées afin de raffiner les modèles.

Cette notion de temporalité rend ce formalisme attrayant pour de nombreuses applications, notamment pour l'apprentissage de réseaux génétiques. Cependant cette richesse de représentation augmente de fait le risque de développer des modélisations à forte complexité.

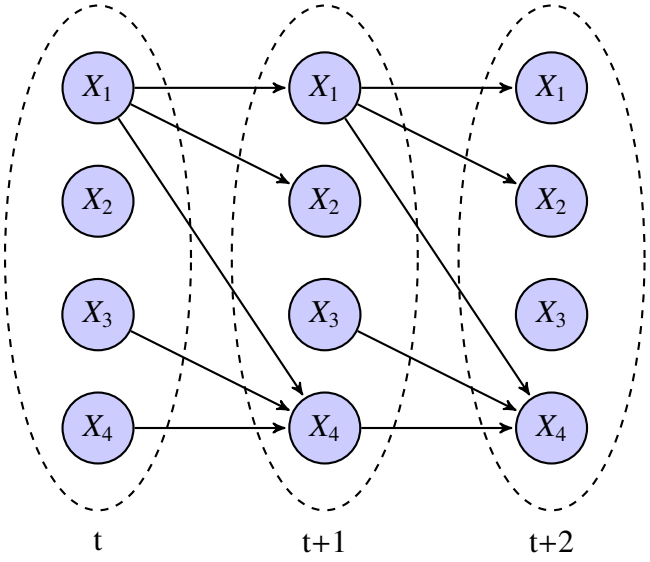


Figure 1.3 – Exemple de réseau bayésien dynamique homogène.

1.3.2 Indépendances et structure du graphe

Nous avons mentionné précédemment le lien entre les indépendances induites par les distributions de probabilités conditionnelles et la structure du graphe. Nous définissons plus en détail ces relations.

1.3.2.1 Condition de Markov

L'hypothèse principale faite par les réseaux bayésiens réside dans la condition de Markov appliquée aux graphes dirigés sans circuit. Cette hypothèse s'exprime comme l'indépendance de toute variable X_i par rapport aux variables qui ne descendent pas d'elle, sachant ses parents dans le graphe. On a donc

$$X_i \perp X_j | Pa(X_i),\ \forall X_j \notin Desc(X_i)$$

1.3.2.2 Séparations dans le graphe

La condition de Markov nous donne ainsi un moyen de définir les indépendances portées par la structure du graphe. Ces indépendances (ou dépendances) peuvent se lire sous la forme d'un flot d'information circulant entre les variables [Pearl, 1988].

Chaque triplet de variables (X_1,X_2,X_3) peut être relié de trois manières différentes dans le graphe :

connexion en série $X_1 \rightarrow X_2 \rightarrow X_3$ (ou $X_1 \leftarrow X_2 \leftarrow X_3$) ;

connexion divergente en X_2 $X_1 \leftarrow X_2 \rightarrow X_3$;

connexion convergente en X_2 $X_1 \rightarrow X_2 \leftarrow X_3$.

Chacune de ces trois configurations exprime une relation d'indépendance entre les trois variables

- $X_1 \rightarrow X_2 \rightarrow X_3 : X_1 \perp X_3 | X_2$ on dit aussi que X_2 bloque la dépendance entre X_1 et X_3 ;
- $X_1 \leftarrow X_2 \rightarrow X_3 : X_1 \perp X_3 | X_2$ on dit de même que X_2 bloque la dépendance entre X_1 et X_3 ;
- $X_1 \rightarrow X_2 \leftarrow X_3 : X_1 \perp X_3 | \emptyset$ et $X_1 \not\perp X_3 | X_i, \forall X_i \in \{X_2 \cup Desc(X_2)\}$ on dit cette fois que X_2 ou l'un de ses descendants active la dépendance entre X_1 et X_3.

Dans le cas d'une connexion en série ou divergente la connaissance de X_2 bloque le flot d'information entre X_1 et X_3 les rendant de fait indépendants. A l'inverse dans le cas d'une connexion convergente, appelée aussi *v-structure*, c'est l'absence de connaissance sur X_2 et sur ses descendants qui bloque ce flot. Lorsque X_2 (ou l'un de ses descendants) est connu la dépendance entre X_1 et X_3 est activée et le flot peut alors circuler. On notera cependant que la configuration $X_1 \rightarrow X_2 \leftarrow X_3$ est une *v-structure* si et seulement si $X_1 \nrightarrow X_3$ et $X_1 \nleftarrow X_3$, dans le cas contraire on parlera de *v-structure couverte*. Cette *v-structure couverte* n'encode alors plus d'indépendance entre X_1 et X_3.

Cette notion de chaîne bloquée ou active permet de définir le principe de *d-séparation* dans le graphe.

D-séparations On dit qu'un ensemble $\mathbb{X}$ de variables est *d-séparé* de $\mathbb{Y}$ par un ensemble $\mathbb{Z}$ dans $\mathcal{G}$ si toute chaîne reliant une variable $x \in \mathbb{X}$ à une variable $y \in \mathbb{Y}$ est bloquée par $\mathbb{Z}$, on note alors $dsep_{\mathcal{G}}(\mathbb{X}, \mathbb{Y}|\mathbb{Z})$ [Pearl, 1988].

Le graphe encode donc par sa structure un ensemble d'indépendances que nous noterons $\mathcal{I}(\mathcal{G})$. La distribution $\mathbb{P}$ encode également un tel ensemble noté $\mathcal{I}(\mathbb{P})$. L'égalité de ces deux ensembles n'est cependant pas immédiate. Nous définissons par la suite les différentes inclusions possibles.

Fidélité Une distribution $\mathbb{P}$ est *fidèle* à un graphe $\mathcal{G}$ si toutes les indépendances de $\mathcal{I}(\mathbb{P})$ sont portées par $\mathcal{G}$ ($\mathcal{I}(\mathbb{P}) \subseteq \mathcal{I}(\mathcal{G})$). En d'autres termes si pour toute indépendance conditionnelle $\mathbb{X} \perp \mathbb{Y} | \mathbb{Z}$ on a $dsep_{\mathcal{G}}(\mathbb{X}, \mathbb{Y}|\mathbb{Z})$.

Carte d'indépendance Nous appelons *carte d'indépendance* de $\mathbb{P}$, tout graphe $\mathcal{G}$ dont l'ensemble des indépendances est porté par $\mathbb{P}$, c'est-à-dire que $\mathcal{I}(\mathcal{G}) \subseteq \mathcal{I}(\mathbb{P})$ [Pearl, 1985].

On notera que si $\mathcal{G}$ est une carte d'indépendance de $\mathbb{P}$ alors cette dernière se factorise sur $\mathcal{G}$. Ce résultat s'obtient à partir de la loi fondamentale généralisée de $\mathbb{P}$

$$\mathbb{P}(X_1, X_2, .., X_p) = \prod_{i=1}^{p} \mathbb{P}(X_i | X_1..X_{i-1})$$

en supposant l'existence d'un ordre sur les variables du graphe $\mathcal{G}$ tel que

$$Pa(X_i) \subseteq \{X_1..X_{i-1}\} \subseteq nonDesc(X_i)$$

Si $\mathcal{G}$ est une carte d'indépendance de $\mathbb{P}$ alors les indépendances de $\mathcal{G}$ qui s'expriment d'après la condition de Markov comme $X_i \perp nonDesc(X_i) | Pa(X_i)$ sont portées par $\mathbb{P}$. Or si $X_i \perp nonDesc(X_i) | Pa(X_i)$ alors $X_i \perp \{X_1..X_{i-1}\} | Pa(X_i)$ dû aux inclusions précédentes.

On peut alors simplifier la loi fondamentale généralisée

$$\mathbb{P}(X_1, X_2, .., X_p) = \prod_{i=1}^{p} \mathbb{P}(X_i | Pa(X_i))$$

retrouvant ainsi la factorisation désirée. La réciproque de ce résultat est également valide.

On remarque par ailleurs que si la distribution $\mathbb{P}$ est fidèle à $\mathcal{G}$ et se factorise sur ce même graphe alors $\mathcal{I}(\mathcal{G}) = \mathcal{I}(\mathbb{P})$. Cependant ces deux conditions ne sont pas vérifiées systématiquement, ainsi certaines distributions se factorisent sur $\mathcal{G}$ sans pour autant lui être fidèle.

Carte d'indépendance minimale Les cartes d'indépendance pour une distribution de probabilité quelconque sont nombreuses. Par exemple les graphes complets qui n'encodent aucune indépendance sont donc de fait des cartes d'indépendance pour toute distribution. Il est donc nécessaire de raffiner cette notion à l'aide des cartes d'indépendance minimales.

Un graphe $\mathcal{G}$ est une carte d'indépendance minimale de $\mathbb{P}$ si $\mathcal{I}(\mathcal{G}) \subseteq \mathcal{I}(\mathbb{P})$ et que pour tout graphe $\mathcal{G}'$ obtenu en enlevant un arc à $\mathcal{G}$ alors $\mathcal{I}(\mathcal{G}') \nsubseteq \mathcal{I}(\mathbb{P})$ [Pearl, 1985].

Malgré cette définition, les cartes d'indépendance minimales ne sont toujours pas uniques. Afin de s'en convaincre nous représentons sur la Figure **1.4** trois cartes d'indépendance minimales possibles pour une distribution $\mathbb{P}$ dont seule la première représente exactement toutes les indépendances de P. Les cartes d'indépendance minimales (b) et (c) possèdent en revanche trop d'arêtes, certaines indépendances portées par $\mathbb{P}$ sont alors perdues telles que $X_1 \perp X_2 | \emptyset$. On dit que (a) représente la *carte parfaite* de $\mathbb{P}$.

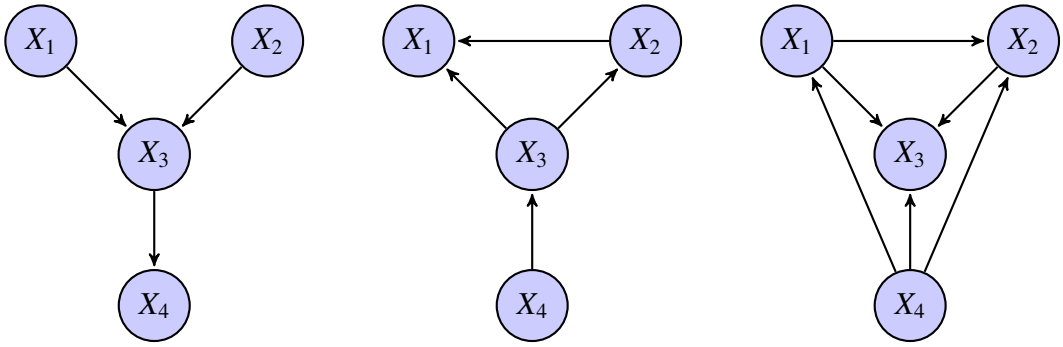


Figure 1.4 – Trois cartes d'indépendance minimales pour une distribution $\mathbb{P}$. Seul le graphe (a) représente toutes les indépendances de $\mathbb{P}$, tandis que (c) est le graphe complet n'encodant aucune indépendance.

Carte parfaite Un graphe $\mathcal{G}$ est une *carte d'indépendance parfaite* de $\mathbb{P}$ si $\mathcal{I}(\mathcal{G}) = \mathcal{I}(\mathbb{P})$ [Pearl, 1985].

Si $\mathbb{P}$ admet une telle carte alors elle est dite *représentable*. Cependant l'existence de cette carte n'est pas assurée. Il est tout de même possible dans ce cas de s'en approcher à l'aide d'une carte d'indépendance minimale qui ne pourra représenter toutes les indépendances de $\mathbb{P}$.

Afin de représenter au mieux $\mathbb{P}$ il convient donc de rechercher sa carte d'indépendance parfaite, ou plutôt l'une de ses cartes car celle-ci n'est pas unique. En effet plusieurs graphes peuvent représenter le même ensemble d'indépendances et sont donc tous des cartes parfaites de $\mathbb{P}$, ces graphes sont ainsi appelés des *équivalents de Markov*.

1.3.3 Équivalent de Markov

Deux graphes $\mathcal{G}_1$ et $\mathcal{G}_2$ sont des *équivalents de Markov* s'ils encodent la même décomposition de la loi jointe, c'est-à-dire si $\mathcal{I}(\mathcal{G}_1) = \mathcal{I}(\mathcal{G}_2)$ [Verma and Pearl, 1990].

Nous avons vu que les *v-structures* induisent des dépendances conditionnelles, ainsi deux graphes possédant les mêmes *v-structures* partagent les mêmes dépendances conditionnelles. De fait une interprétation graphique de l'équivalence de Markov consiste à déclarer deux graphes $\mathcal{G}_1$ et $\mathcal{G}_2$ équivalents si et seulement si ils partagent le même squelette et les mêmes *v-structures*.

Représentation graphique Afin de représenter l'ensemble des DAG équivalents de Markov à l'aide d'un unique graphe, on a recourt aux graphes partiellement dirigés appelés pDAG (pour *partial Directed Acyclic Graph*).

Pour un DAG initial, on obtient le pDAG représentant sa classe d'équivalence en conservant les arcs dits *non-réversibles* et en transformant les arcs *réversibles* en arêtes. Le pDAG obtenu est alors

appelé le graphe *essentiel*, ce graphe unique, représente à lui-seul l'ensemble des DAG appartenant à une même classe d'équivalence de Markov.

Un arc est dit *réversible* s'il n'intervient pas dans une *v-structure* et s'il ne crée ni circuit ni nouvelle *v-structure* lorsque son orientation est changée.

Si chaque classe d'équivalence est représentable par un pDAG, tout pDAG ne représente pas pour autant une classe d'équivalence. Pour certains d'entre-eux, chaque orientation possible des arêtes mène obligatoirement à la création de *v-structure* (ou de circuit) et donc à des DAG non-équivalents comme présentés sur la Figure **1.5**. Lorsque l'orientation des arêtes du pDAG permet de définir un ensemble non vide de DAG équivalents, on dit alors que le pDAG est *instanciable*. Un pDAG est donc *instanciable* s'il est le représentant d'une unique classe d'équivalence de Markov, on utilise également dans ce cas le terme de cpDAG (pour *completed pDAG*).

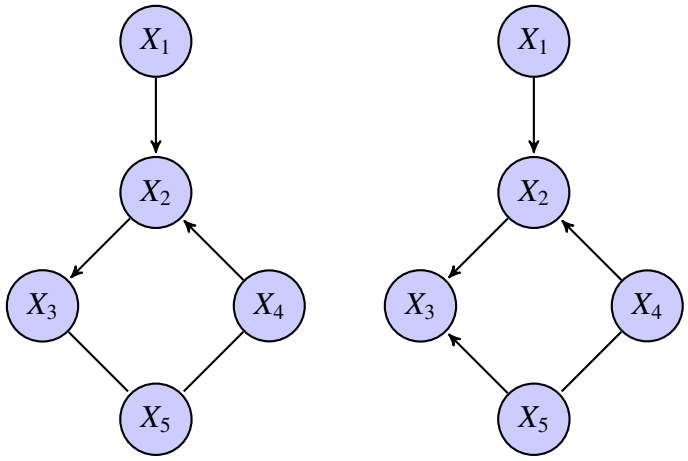


Figure 1.5 – (a) Exemple de pDAG non-instanciable, chacun des 4 couples d'orientations possibles pour $X_3 - X_5$ et $X_4 - X_5$ crée soit un circuit, soit une *v-structure*. (b) Exemple de cpDAG instanciable où les deux orientations possibles pour $X_4 - X_5$ mènent à la même classe d'équivalence.

1.3.4 Causalité dans les réseaux bayésiens

Les réseaux bayésiens permettent de représenter le fonctionnement de systèmes réels et notamment les liens causaux entre les variables du graphe. A l'inverse interpréter en terme de causalité les arcs appris à partir de données n'est pas immédiat. Nous venons notamment de voir le cas des réseaux équivalents de Markov dont certains arcs ne possèdent pas d'orientation forte, ceux-ci peuvent alors être inversés sans changer l'ensemble des indépendances induites. Dans ce cas, seuls les arcs non-inversibles du graphe possèdent un réel sens causal.

L'une des solutions consiste alors à utiliser des données dites d'*intervention*, les variations observées pour les différentes variables lors d'une perturbation ciblée sur l'une d'entre elles, permettent ainsi d'orienter de manière certaine ces arcs. Certaines approches ont été développées afin de déterminer la suite de perturbations à appliquer au réseau permettant de retrouver un maximum de ces relations causales [Tong and Koller, 2001, Pournara and Wernisch, 2004, Hauser and Bühlmann, 2012], ces approches effectuent alors un apprentissage dit *actif*. Cependant, dans cette thèse nous ne nous attachons pas à cet objectif et restons dans le cas d'un apprentissage à partir de données d'observation uniquement. De plus dans la suite de ce manuscrit, nous présenterons des comparatifs effectués à partir de réseaux rendus non-orientés, évitant ainsi ce biais d'interprétation. Nous verrons malgré tout que dans le cas de données génétiques, l'utilisation de différentes sources d'information permet d'orienter fortement certaines relations sans avoir recourt à des perturbations expérimentales.

Pearl and Verma [1991] introduisent alors des RB spécifiques, appelés *réseaux causaux*, dont chaque arc représente une relation purement causale. Différentes méthodes visent ainsi à apprendre la structure de ce type de réseaux [Meganck et al., 2006, Leray et al., 2008].

1.3.5 Énumération des réseaux bayésiens

Robinson [1977] a défini le nombre de DAG possibles suivant le nombre p de variables aléatoires grâce à la formule de récurrence suivante

$$N(p) = \sum_{i=1}^{p} (-1)^{i+1} C_p^i 2^{i(p-1)} N(p-i)$$

avec C_p^i le coefficient binomial donnant le nombre de combinaisons de i éléments parmi p.

Ce nombre croit de manière super-exponentielle ce qui laisse entrevoir la difficulté d'apprendre la structure de ces graphes.

1.3.6 Apprentissage des réseaux bayésiens

L'apprentissage d'un réseau bayésien sachant un ensemble d'observations peut se décomposer en deux étapes, la première consiste à apprendre sa structure, c'est-à-dire le graphe associé au réseau. Une fois sa structure connue, il est nécessaire d'estimer ses distributions de probabilités conditionnelles. Nous nous restreignons ici au cas des réseaux bayésiens discrets où les distributions de probabilités sont décrites par un ensemble de paramètres. De plus, nous supposons que les observations sont complètes, nous ne traiterons donc pas de la situation où des données sont manquantes.

1.3.6.1 Apprentissage de la structure

Apprendre la structure d'un réseau bayésien consiste à rechercher la carte parfaite des distributions de probabilités conditionnelles sous-jacentes aux observations, ou, lorsque cette dernière n'est pas représentable, de rechercher une de ses cartes d'indépendance minimales. Pour cela plusieurs

approches sont employées que ce soit par la recherche directe des indépendances entre variables, ou par l'exploration de l'espace des structures possibles. De manière générale cet apprentissage est complexe dû au nombre élevé de réseaux bayésiens possibles comme nous l'avons mentionné. Nous présentons dans la section 2.1 différentes approches permettant d'apprendre au mieux cette structure.

1.3.6.2 Apprentissage des paramètres

Dans le cas de réseaux bayésiens discrets, c'est-à-dire lorsque chaque variable X_i prend ses valeurs dans un ensemble fini de taille D_{X_i}, chaque distribution de probabilités conditionnelles $\mathbb{P}(X_i|Pa(X_i))$ se définit suivant un ensemble de paramètres θ_i. Ces paramètres peuvent être représentés sous la forme d'une matrice dont chaque ligne k correspond à la $k^{\text{ème}}$ valeur de X_i et chaque colonne j, à la $j^{\text{ème}}$ configuration des valeurs de $Pa(X_i)$. La cellule correspondante contient alors le paramètre

$$\theta_{ijk} = \mathbb{P}(X_i = k|Pa(X_i) = j)$$

Ces paramètres peuvent être estimés de différentes manières. L'une des approches les plus courantes consiste à utiliser le maximum de vraisemblance (MLE) qui ne suppose aucun *a priori*. D'autres approches posent au contraire un *a priori* sur la distribution des paramètres, nous verrons par exemple le cas où cette distribution suit une loi de Dirichlet.

Par maximum de vraisemblance L'estimation par MLE des paramètres $\hat{\theta}_i$ d'une variable X_i sachant son ensemble de parents $Pa(X_i)$ s'effectue par comptage dans la table d'occurrence.

$$\hat{\theta}_{ijk} = \frac{[X_i = k, Pa(X_i) = j]}{[Pa(X_i) = j]}$$

où $[X_i = k, Pa(X_i) = j]$ représente le nombre d'observations pour lesquelles X_i prend sa $k^{\text{ème}}$ valeur et l'ensemble $Pa(X_i)$ est dans sa $j^{\text{ème}}$ configuration.

***A priori* de Dirichlet** Supposer un *a priori* de Dirichlet sur les paramètres θ_{ijk} tel que $\mathbb{P}(\theta_{ijk}) = Dir(\theta_{ijk}|\alpha_{ijk})$ nécessite de fixer l'ensemble des coefficients α_{ijk} de la distribution de Dirichlet. Ces paramètres peuvent être vus comme des observations imaginaires de chaque configuration possible permettant ainsi de lisser les variations entre les paramètres θ_{ijk}. Prenons l'exemple d'une configuration particulière rare ($X_i = k$ et $Pa(X_i) = j$) qui peut, lorsque le nombre d'échantillons est faible, ne pas être observée, ainsi au lieu de fixer $\hat{\theta}_{ijk} = 0$, ce paramètre prend une valeur dépendante de l'*a priori* α_{ijk}.

L'estimation bayésienne permet ainsi de définir les paramètres $\hat{\theta}_{ijk}$ les plus probables sachant cette distribution de Dirichlet, on parle alors d'estimation par maximum *a posteriori* (MAP).

$$\hat{\theta}_{ijk} = \frac{[X_i = k, Pa(X_i) = j] + \alpha_{ijk} - 1}{[Pa(X_i) = j] + \alpha_{ij} - 1} \text{ avec } \alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$$

Nous verrons dans la section 2.1.2, l'utilisation faite de ces deux estimateurs afin de définir des scores permettant d'évaluer la qualité d'un réseau bayésien.

Chapitre 2

État de l'art

Sommaire

2.1 Apprentissage de la structure d'un réseau bayésien 26
2.1.1 Recherche d'indépendances conditionnelles 27
2.1.1.1 Mesures de corrélation 27
2.1.1.2 Algorithmes de recherche d'indépendances 29
2.1.2 Optimisation d'un score 30
2.1.2.1 Fonctions de score 30
2.1.2.2 Méthodes optimales sous contraintes 33
2.1.2.3 Espace de recherche des DAG 34
2.1.2.4 Espace de recherche des cpDAG 39
2.1.2.5 Espace de recherche des ordres 41
2.1.3 Approches hybrides 42
2.2 Discrétisation des données 46
2.2.1 Méthodes de discrétisations 46
2.2.1.1 Méthodes univariées 46
2.2.1.2 Méthodes multivariées 48
2.2.2 Choix du nombre de classes 49
2.3 Conclusion 50

Ce premier chapitre de l'état de l'art expose dans un premier temps les différentes approches développées afin d'apprendre la structure d'un réseau bayésien discret (noté RB). Nous distinguons trois classes d'approches. La première consiste à rechercher directement les indépendances présentes dans les observations afin de construire le graphe correspondant à ces indépendances. La seconde représente les approches utilisant un score afin de guider le parcours dans l'espace des graphes possibles. Nous décrivons à cette occasion différents scores couramment employés ainsi que certaines de leurs propriétés. La dernière classe utilise les deux stratégies précédentes de concert afin de proposer des méthodes hybrides. Nous abordons également dans la seconde partie de ce chapitre, le problème de la discrétisation, souvent nécessaire dans le cas de données réelles. Nous présentons

quelques méthodes classiques de discrétisation ainsi qu'une approche supervisée visant à réduire la perte d'information mutuelle entre paires de variables lors de la discrétisation.

2.1 Apprentissage de la structure d'un réseau bayésien

Complexité de la tache d'apprentissage Différents facteurs peuvent venir perturber l'apprentissage de la structure du RB, le plus fréquent d'entre eux est la présence de données non observées, on parle alors de données manquantes. Dans ce cas deux approches sont possibles, la première consiste à compléter les données, avant l'apprentissage, par une inférence statistique comme utilisée en section 8.3. La deuxième solution réside dans l'utilisation d'algorithmes d'apprentissage gérant la présence de données manquantes.

Une autre difficulté réside dans l'existence de variables latentes. Une variable latente est une variable dont dépend une ou plusieurs variables observées sans être elle même observée. La modélisation de phénomènes biologiques tel la régulation entre gènes, n'échappe pas à la présence de variables latentes.

La dernière difficulté provient de la complexité intrinsèque de l'apprentissage de la structure d'un DAG. Nous avons vu dans le premier chapitre que le nombre de DAG possibles augmente de façon super-exponentielle en fonction du nombre de variables. Bien que la dimension de cet espace de recherche laisse entrevoir la difficulté d'apprendre le vrai réseau, elle n'en est pas une preuve pour autant. Chickering et al. [2004] apportent une preuve du caractère NP-dur de l'apprentissage du RB optimal en terme de score (que nous définirons en section 2.1.2.1), en réduisant un problème connu pour être lui même NP-dur à ce problème d'apprentissage. Cette réduction permet de qualifier l'apprentissage de la structure optimale comme étant au moins aussi dur à résoudre qu'un problème NP-dur. Ainsi le problème de référence utilisé est la recherche pour un graphe orienté, d'un sous-ensemble borné d'arcs contenant au moins un arc de chaque circuit présent dans ce graphe [Festa et al., 2009]. La preuve apportée par Chickering et al. [2004] est valide lorsque le nombre d'observations est élevé et que ces observations sont fidèles à une loi de probabilité représentable par un DAG dès lors que le nombre de parents est supérieur à 2.

Approches pour l'apprentissage Depuis plus de 20 ans de nombreuses stratégies ont été mises en place afin d'approcher au plus près la solution optimale, celles-ci sont généralement classées en deux catégories.

Recherche d'indépendances L'approche la plus naturelle consiste à rechercher toutes les relations d'indépendances entre les variables et d'obtenir ainsi les dépendances directes et donc la structure du réseau. Différentes mesures d'indépendance et stratégies ont été déployées dans cet objectif, celles-ci seront développées dans la section 2.1.1.

Maximisation d'un score L'utilisation d'un score mesurant la vraisemblance du modèle sachant les observations représente la seconde approche. Dans ce cas les heuristiques développées s'attachent à rechercher le réseau maximisant ce score. La description de ces heuristiques ainsi que différents scores fera l'objet de la section 2.1.2.

Une troisième catégorie de méthodes dites *"hybrides"* ou *"mixtes"* combinent les deux approches précédentes. Cette hybridation peut s'effectuer à différents niveaux comme nous le soulignerons en section 2.1.3.

Nous présenterons dans la suite de ce chapitre ces différentes approches d'apprentissage de la structure en se restreignant aux méthodes qui supposent que la base d'observations ne contient pas de données manquantes et que le processus ayant généré ces données (qu'il soit artificiel ou non) ne contient pas de variables latentes.

2.1.1 Recherche d'indépendances conditionnelles

2.1.1.1 Mesures de corrélation

Nous présentons dans cette section deux mesures employées afin de mesurer le degré d'indépendance de deux variables. Bien que ces deux mesures soient parmi les plus courantes, d'autres mesures peuvent être employées comme le rapport de vraisemblance ($\mathcal{G}^2$), la mesure de corrélation de Pearson ou le test de Fisher.

Test d'indépendance du Chi-2 Le test d'indépendance du chi-2 (noté χ^2) [Chernoff and Lehmann, 1954] permet de déterminer si deux variables aléatoires X et Y sont indépendantes étant donné un niveau de risque de première espèce. On définit pour cela l'hypothèse nulle H_0 qui postule que ces deux variables sont indépendantes. L'objectif du test est donc d'accepter ou de rejeter H_0, en prenant le risque de se tromper sous H_0 de α.
En pratique soit X et Y deux variables aléatoires de domaines respectifs D_X et D_Y , on mesure la valeur du χ^2 pour ces deux variables :

$$\chi^2 = \sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \frac{([O_{ij}] - [E_{ij}])^2}{[E_{ij}]}$$

où $[O_{ij}]$ représente le nombre réel d'observations pour lesquelles X prend sa i$^{\text{ème}}$ valeur et Y sa j$^{\text{ème}}$ valeur tandis que $[E_{ij}]$ représente le nombre attendu si H_0 est vérifiée (indépendance entre X et Y).
Afin d'accepter ou de rejeter H_0, la valeur calculée du test χ^2 est comparée à la loi du χ^2 de degré de liberté k. Dans le cas d'une comparaison entre deux variables X et Y de domaines respectifs D_X et D_Y, le nombre de degrés de liberté est $k = (D_X - 1)\,(D_Y - 1)$.

On obtient la procédure de décision suivante :

$$\begin{cases} \chi^2 < q_{\chi^2_k}^{1-\alpha} & \rightarrow \quad H_0 \; est \; acceptée \\ \chi^2 >= q_{\chi^2_k}^{1-\alpha} & \rightarrow \quad H_0 \; est \; rejetée \end{cases}$$

où $q^{1-\alpha}_{\chi^2_k}$ dénote le quantile à $(1-\alpha)\%$ de la loi du χ^2 à k degrés de liberté.

La valeur α généralement fixée à 0.05 permet de régler la stringence du test, une valeur proche de 0 fera que H_0 sera acceptée plus facilement, à l'inverse en augmentant α le test rejettera H_0 plus fréquemment.

Le test du χ^2 peut s'étendre au cas conditionnel, dans ce cas on a

$$\chi^2 = \sum_{k=1}^{D_Z} \sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \frac{([O_{ijk}] - [E_{ijk}])^2}{[E_{ijk}]}$$

où $[O_{ijk}]$ représente le nombre réel d'observations pour lesquelles X prend sa i$^{\text{ème}}$ valeur,Y sa j$^{\text{ème}}$ valeur et la variable de conditionnement Z sa k$^{\text{ème}}$ valeur tandis que $[E_{ijk}]$ représente le nombre attendu si H_0 est vérifiée (indépendance entre X et Y sachant Z).

Information mutuelle conditionnelle La mesure d'information mutuelle entre deux variables aléatoires X et Y définit la quantité d'information partagée entre ces deux variables, intuitivement deux variables dépendantes possèdent une information mutuelle élevée tandis qu'une information mutuelle nulle révèle l'indépendance. L'origine de cette mesure provient des premières notions de représentations de l'information en informatique. Le théorème de Shannon [Shannon, 1948] pose la question de l'espace mémoire nécessaire (en bits) pour stocker l'information d'une variable X, en d'autres termes cela revient à connaître le contenu en information de X. Il définit cette quantité comme tendant vers la valeur de l'entropie de X

$$H(X) = -\sum_{i=1}^{D_X} \mathbb{P}(X = i) \log_2(\mathbb{P}(X = i))$$

où $\mathbb{P}(X = i)$ représente la probabilité d'observer la i$^{\text{ème}}$ valeur du domaine D_X de X.
De même on définit l'entropie conjointe $H(X, Y)$ mesurant la quantité d'information contenue dans X et Y.

$$H(X, Y) = -\sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \mathbb{P}(X = i, Y = j) \log_2(\mathbb{P}(X = i, Y = j))$$

Ainsi l'information mutuelle entre X et Y, notée $IM(X, Y)$, peut se définir par :

$$IM(X, Y) = H(X) + H(Y) - H(X, Y)$$

On remarque que dans le cas où X et Y sont indépendants, c'est-à-dire qu'elles ne possèdent aucune information en commun, l'entropie conjointe est la somme exacte des entropies de X et Y soit $H(X, Y) = H(X) + H(Y)$ ce qui implique bien $IM(X, Y) = 0$.
De manière plus générale on note :

$$IM(X, Y) = \sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \mathbb{P}(X = i, Y = j) \log_2(\frac{\mathbb{P}(X = i, Y = j)}{\mathbb{P}(X = i), \mathbb{P}(Y = j))})$$

L'information mutuelle conditionnelle s'obtient simplement à partir de cette définition comme étant

$$IM(X,Y|Z) = \sum_{k=1}^{D_Z} \mathbb{P}(Z=k) \sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \mathbb{P}(X=i, Y=j|Z=k) \log_2 \frac{\mathbb{P}(X=i, Y=j|Z=k)}{\mathbb{P}(X=i|Z=k), \mathbb{P}(Y=j|Z=k))}$$

où Z représente la variable de conditionnement de domaine D_Z.
A la différence du test du χ^2, l'information mutuelle telle que définie précédemment ne répond pas à la question : "X et Y sont elles indépendantes ?". Cette mesure indique uniquement leur degré de dépendance (ou d'indépendance), il convient donc de fixer un seuil pour pouvoir l'utiliser comme test d'indépendance.

2.1.1.2 Algorithmes de recherche d'indépendances

PC L'algorithme PC [Spirtes et al., 1993] utilise un test statistique tel le test du χ^2 pour évaluer s'il y a indépendance conditionnelle entre deux variables. Pour cela la méthode débute par un graphe complet et étudie un à un tous les arcs de ce graphe, pour chacun d'eux, il teste dans un premier temps l'indépendance *a priori* (sans aucune variable de conditionnement) afin d'éliminer une première vague d'arcs entre deux nœuds déclarés indépendants. Puis il réexamine tous les arcs restant en conditionnant successivement le test d'indépendance avec chaque voisin d'un des deux nœuds, permettant ainsi d'éliminer d'autres relations. L'algorithme continue en augmentant progressivement la taille de l'ensemble de conditionnement, tant que le nombre de voisins des nœuds considérés le permet. Le graphe obtenu après cet étape est un graphe non orienté dû au caractère symétrique du test employé, s'en suit alors une orientation partielle du graphe en déterminant les *v-structures* à partir des indépendances conditionnelles ce qui permet également de déterminer par propagation d'autres orientations afin d'obtenir un graphe partiellement orienté.

BNPC La méthode BNPC [Cheng et al., 2002] (ou TPDA) constitue une évolution de l'algorithme PC reposant sur le même principe de fonctionnement avec toute fois un comportement moins unisens, elle se décompose en trois phases. La première consiste à calculer la valeur de l'information mutuelle entre chaque paire de variables et ne conserver que les couples pour lesquels cette valeur est supérieure à un seuil fixé. Ces couples sont ensuite classés suivant cette même mesure, constituant ainsi la liste triée sur laquelle travaille l'algorithme dans les phases suivantes. En appliquant l'algorithme MWST défini en section 2.1.2.2 sur cette liste d'arcs potentiels, pondérés par l'information mutuelle, l'algorithme définit une première structure arborescente. Une fois cet arbre établi la 2ème phase de BNPC considère successivement les arcs de la liste non ajoutés jusque là. L'arc est ajouté si et seulement si la structure courante du graphe ne permet pas d'invalider la dépendance associée. Lorsque tous les arcs de la liste sont traités une 3ème phase semblable à l'algorithme PC est appliquée. Cette étape consiste à opérer pour chaque arc dans le sens inverse de la liste initiale, un test de séparation pour regarder si la nouvelle structure du graphe révèle certaines indépendances. De manière similaire à l'algorithme PC on obtient après une phase d'orientation des *v-structures* et d'une propagation sur d'autres arêtes, une orientation partielle du graphe.

2.1.2 Optimisation d'un score

L'utilisation d'un score basé sur la vraisemblance du modèle offre une alternative intéressante à l'utilisation de tests statistiques d'indépendance. Ce score permet d'estimer la qualité d'un réseau dans son intégralité au vu des observations, à la différence des tests d'indépendance s'effectuant à une échelle locale. Le problème d'apprentissage se transforme alors en un problème d'optimisation de ce score dans un espace des DAG de taille super-exponentielle. Deux questions sont alors posées, la première porte sur la définition du score qui doit permettre de tendre asymptotiquement vers la vraie structure tout en restant peu coûteux à évaluer. La seconde provient de la taille de l'espace de recherche qui nécessite de définir une heuristique guidant la recherche au travers de cet espace.

2.1.2.1 Fonctions de score

Caractéristiques d'un score Deux caractéristiques sont souvent souhaitées pour l'utilisation d'un score, le premier concerne la *décomposabilité* de celui-ci afin de limiter le coût de l'évaluation du RB tandis que l'*équivalence* assure une certaine cohérence au vu de la définition d'un RB.

Décomposabilité Malgré le développement d'heuristiques tachant de guider rapidement la recherche vers une structure solution, le nombre de structures différentes à considérer reste important. Il est donc primordial de limiter le coût d'évaluation d'une structure donnée. La majeur partie des heuristiques de recherche agissent par modification locale du graphe, typiquement l'ajout ou la suppression d'un seul arc au graphe courant. Dans ce cas, si le score est décomposable, il est inutile de recalculer le score du graphe tout entier mais uniquement de déterminer la variation locale du score induit par cette modification, permettant ainsi de réduire la complexité des heuristiques de recherche. Un score S est dit décomposable si il peut s'écrire sous la forme d'une somme de scores locaux pour chacune des p variables qui ne dépend que de l'état de ses parents dans le graphe.

$$S(\mathcal{G}) = \sum_{i=1}^{p} s(X_i|U_i)$$

où $s(X_i|U)$ représente le score local de X_i sachant l'ensemble de ses parents U_i défini par le graphe $\mathcal{G}$.

Equivalence Un RB représente avant tout un ensemble d'indépendances conditionnelles, cependant plusieurs structures peuvent encoder les mêmes indépendances comme présenté en section 1.3.3. Il est donc important d'attribuer à chacune des structures d'une même classe d'équivalence le même score. Les scores vérifiant cette condition sont dit "équivalents". A la différence du principe de décomposabilité vivement souhaité dans le cas de RB de grande taille, l'équivalence du score n'est pas indispensable. Cependant l'utilisation d'un score non-équivalent nécessite une réflexion sur le sens à donner à une structure préférée parmi une classe d'équivalence. Nous abordons par la suite différents scores tous décomposables qui ne sont cependant pas nécessairement équivalents.

Scores existants Chacun des scores doit maximiser la probabilité $\mathbb{P}(\mathcal{G}|\mathbf{D})$ du graphe $\mathcal{G}$ sachant les observations $\mathbf{D}$. Or d'après la formule de Bayes

$$\mathbb{P}(\mathcal{G}|\mathbf{D}) = \frac{\mathbb{P}(\mathbf{D}|\mathcal{G})\ \mathbb{P}(\mathcal{G})}{\mathbb{P}(\mathbf{D})}$$

avec $\mathbb{P}(\mathbf{D}|\mathcal{G})$ la vraisemblance marginale des données sachant le modèle, $\mathbb{P}(\mathcal{G})$ l'*a priori* sur le modèle et $\mathbb{P}(\mathbf{D})$ l'*a priori* sur les données qui ne dépend pas du modèle et que l'on peut par conséquent ignorer.

Afin de ne privilégier aucune structure *a priori*, $\mathbb{P}(\mathcal{G})$ est généralement assimilée à une constante pour l'ensemble des graphes, revenant ainsi à sélectionner le graphe maximisant $\mathbb{P}(\mathbf{D}|\mathcal{G})$. Nous verrons en section 4.3 une hypothèse faite sur $\mathbb{P}(\mathcal{G})$ vue comme un biais touchant l'*a priori* sur la connectivité du graphe appris.

$\mathbb{P}(\mathbf{D}|\mathcal{G})$ représente la vraisemblance des données pour un graphe considéré. Lorsque les paramètres θ sont connus, il est possible de calculer cette vraisemblance :

$$\mathbb{P}(\mathbf{D}|\mathcal{G},\tilde{\theta}) = \prod_{i=1}^{p}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}\tilde{\theta}_{ijk}^{\,[X_i=k,Pa(X_i)=j]} \tag{2.1}$$

Cependant, ces paramètres sont rarement connus à l'avance, la vraisemblance se calcule alors par intégration sur les paramètres θ :

$$\mathbb{P}(\mathbf{D}|\mathcal{G}) = \int_{\theta}\mathbb{P}(\mathbf{D}|\mathcal{G},\theta)\mathbb{P}(\theta|\mathcal{G})d\theta \tag{2.2}$$

Cette intégrale n'étant pas calculable dans le cas général, plusieurs approches d'estimation ont été proposées, chacune d'elles donnant naissance aux scores présentés par la suite.

BIC Schwarz [1978] utilise l'approximation de Laplace pour estimer l'intégrale (2.2) . L'expression ainsi obtenue se simplifie sous l'hypothèse d'un grand nombre d'observations, définissant le score *Bayesian Information Criterion* (*BIC*).

$$BIC(\mathcal{G}) = \log(\mathbb{P}(\mathbf{D}|\mathcal{G},\hat{\theta})) - \frac{1}{2}\log(N)Dim(\mathcal{B}) \approx \log(\mathbb{P}(\mathbf{D}|\mathcal{G}))$$

où $\mathcal{B} = (\mathcal{G},\hat{\theta})$ représente le réseau bayésien constitué du graphe $\mathcal{G}$ et des paramètres $\hat{\theta}$ estimés par maximum de vraisemblance à partir des N observations. Ce score se décompose en un premier terme de la log-vraisemblance des données par rapport au modèle et d'un second terme qui pénalise les structures complexes. Le score *BIC* est par ailleurs score-équivalent.

BD Plutôt que d'effectuer une approximation de la vraisemblance Cooper and Hersovits [1992] considèrent les paramètres $\theta_{\mathbf{ij}}$ comme indépendants et suivant une loi de Dirichlet d'hyper-paramètres $\alpha_{\mathbf{ijk}}$. Sous ces hypothèses l'intégrale (2.2) s'exprime sous la forme du score *Bayesian Dirichlet* (*BD*).

$$BD(\mathcal{G}) = \mathbb{P}(\mathbf{D}|\mathcal{G}) = \prod_{i=1}^{p}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(n_{ij}+\alpha_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(n_{ijk}+\alpha_{ijk})}{\Gamma(\alpha_{ijk})}$$

où r_i représente la taille du domaine de la variable X_i, $q_i = \prod_{X_j \in Pa(X_i)} r_j$, le nombre de configurations possibles pour les parents de X_i, n_{ijk}, le nombre d'occurrences de la configuration $(X_i = k, Pa(X_i) = j)$ sur les N observations ainsi que l'hyper-paramètre α_{ijk} associé, avec $n_{ij} = \sum_{k=1}^{r_i} n_{ijk}$ et $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$.

Comme nous l'avons vu dans le Chapitre 1, chaque hyper-paramètre α_{ijk} représente un nombre d'observations imaginaires où l'on a $X_i = k$ et $Pa(X_i) = j$, servant ainsi d'*a priori* sur les observations. Tel que défini par Cooper and Hersovits [1992], le score *BD* requiert de la part de l'utilisateur une valeur pour l'ensemble des hyper-paramètres, ce qui se révèle impossible en pratique. Une des possibilité consiste à attribuer à chacun d'eux une même constante, typiquement $\alpha_{ijk} = 1 \quad \forall i, j, k$ connu sous le nom d'*a priori* K2. Cependant cet *a priori* suppose que le nombre d'observations imaginaires varie en fonction des structures considérées. Une structure plus dense nécessite de définir plus d'hyper-paramètres et suppose donc davantage d'observations *a priori*. De plus l'*a priori* K2 n'est pas équivalent.
Pour pallier cette incohérence Heckerman et al. [1995] définit le score BD_e où chaque α_{ijk} dépend de la probabilité d'observer $X_i = k$ et $Pa(X_i) = j$ on a ainsi

$$\alpha_{ijk} = \alpha \, \mathbb{P}(X_i = k, Pa(X_i) = j)$$

où α représente le nombre d'observations imaginaires, valeur fixe quelle que soit la structure considérée. Le score BD_e possède en outre l'avantage d'être équivalent (d'où son nom '*Bayesian Dirichlet equivalent*'). Cependant il est nécessaire de définir la probabilité *a priori* $\mathbb{P}(X_i = k, Pa(X_i) = j)$, comme souvent en l'absence de connaissance cette probabilité peut être choisie comme uniforme. On obtient alors

$$\mathbb{P}(X_i = k, Pa(X_i) = j) = \frac{1}{r_i q_i}$$

d'où

$$\alpha_{ijk} = \frac{\alpha}{r_i q_i}$$

Dans ce cas on parle du score *BDeu* (pour '*Bayesian Dirichlet equivalent uniform*') où α devient l'unique paramètre de ce score. Malheureusement ce score se révèle très sensible au choix de ce paramètre [Silander et al., 2007].

fNML Silander et al. [2010] proposent une approche basée sur le principe du maximum de vraisemblance normalisé (*Normalized Maximum Likelihood* (NML)) [Shtarkov, 1987].

$$NML(\mathcal{G}) = \frac{\mathbb{P}(\mathbf{D}|\mathcal{G}, \hat{\theta})}{\sum_{\mathbf{D}'} \mathbb{P}(\mathbf{D}'|\mathcal{G}, \hat{\theta})}$$

où la normalisation s'effectue sur la totalité des ensembles d'observations possibles de même taille. L'idée sous-jacente est de sélectionner la structure qui représente au mieux les observations et uniquement celles-ci afin de ne pas retenir une structure consensus qui représenterait un grand nombre d'ensembles d'observations possibles. Bien que le cardinal de cet ensemble soit exponentiel, Kontkanen and Myllymäki [2007] définissent une procédure linéaire en temps afin d'effectuer un

calcul local pour chaque variable multinomiale $X_i \in \mathbf{X}$. Silander et al. [2010] utilisent ce résultat pour définir le score *factorized Normalized Maximum Likelihood* ($fNML$).

$$fNML(\mathcal{G}) = \log(\mathbb{P}(\mathbf{D}|\mathcal{G}, \hat{\theta})) - \sum_{i=1}^{p} \sum_{j=1}^{q_i} \log(C_{n_{ij}}^{r_i})$$

avec $C_{n_{ij}}^{r_i}$ le coefficient de normalisation pour la variable X_i [Kontkanen and Myllymäki, 2007].
Ce score s'exprime de façon similaire au critère BIC, avec un terme de pénalité qui dans ce cas ne dépend pas uniquement de la dimension du graphe mais aussi des données à l'instar de *BDeu* sans avoir de paramètre à régler. Par ailleurs les scores BIC et $fNML$ se révèlent être asymptotiquement équivalents [Silander et al., 2010], cependant à la différence de BIC, $fNML$ n'est pas score-équivalent.

AIC Le score AIC définit par Akaike [1974] est similaire dans la formulation au score BIC mais diverge dans le raisonnement qui lui donna naissance. L'idée étant ici de sélectionner le RB dont la densité de probabilité est la plus proche du modèle ayant servit à générer les observations tout en s'assurant de la parcimonie de la structure (afin d'éviter le sur apprentissage). C'est dans cette optique de compromis entre un modèle explicatif mais simple que se définit ce score :

$$AIC(\mathcal{G}) = 2\log(\mathbb{P}(\mathbf{D}|\mathcal{G}, \hat{\theta})) - 2Dim(\mathcal{B})$$

MDL L'idée du score MDL proposée par Rissanen [1978] puis développée par Bouckaert [1993] suit la même idée que le score AIC avec une vision plus informatique en comparant le problème de sélection de modèle à celui de compression de données. Ainsi il convient de chercher le modèle le plus compact et qui permet une fois celui-ci connu de représenter les données de la manière la plus réduite, autrement dit, sélectionner le modèle qui représente au mieux les données.

$$MDL(\mathcal{G}) = \log(\mathbb{P}(\mathbf{D}|\mathcal{G}, \hat{\theta})) - \frac{1}{2}\log(N)Dim(\mathcal{B})$$

Le premier terme de l'expression représente l'espace nécessaire à la représentation des données sachant le modèle, tandis que le second mesure l'espace nécessaire pour coder le modèle. On remarque immédiatement que la formulation de ce score correspond au score BIC. Le score MDL peut cependant s'exprimer sous différentes formes en fonction de la représentation adoptée pour décrire le modèle (le second terme de l'expression du score). Ainsi Lam and Bacchus [1993] et Suzuki [1999] conçoivent d'autres formulations pour définir le modèle menant à des variantes de l'expression présentée ci dessus.

2.1.2.2 Méthodes optimales sous contraintes

Nous avons vu précédemment que la recherche du graphe optimal en terme de score est un problème NP-dur dans le cas général, cependant certaines approches optimales assurent de trouver le RB ayant le score le plus élevé sous certaines contraintes. Ces contraintes peuvent porter soit sur le nombre de variables du réseau considéré soit sur l'espace des structures parcourues.

Restriction à des petits réseaux L'énumération exhaustive des structures devient rapidement impossible pour des réseaux dont le nombre de variable dépasse la dizaine. Cependant différentes techniques visent à repousser progressivement cette limite. de Campos et al. [2009] cherchent dans un premier temps pour chaque variable son ensemble optimal de parents sans tenir compte des circuits puis utilisent un algorithme de Branch and Bound afin de satisfaire l'acyclicité de la structure. Yuan et al. [2011] utilisent la programmation dynamique associée à une reformulation du problème d'apprentissage de la structure en une recherche de chemin minimum dans un graphe. L'utilisation de la programmation linéaire offre également des solutions comme le montre Jaakkola et al. [2010]. Cependant ces différentes approches permettent seulement de traiter des réseaux comprenant moins de 50 variables dans un temps raisonnable. De plus les espaces mémoires nécessaires pour traiter ces réseaux restent bien souvent prohibitifs.

Restriction de l'espace de recherche Une autre approche consiste à réduire l'espace des structures possibles, en considérant par exemple l'espace des arbres dont la taille est de p^{p-2} pour p variables.

MWST Apprendre un RB dont la structure est celle d'un arbre se rapproche de la recherche d'un arbre couvrant de poids maximum (MWST pour Maximum Weighted Spanning Tree). Chow and Liu [1968] propose pour cela d'utiliser l'information mutuelle entre chaque paire de variable afin de pondérer l'arête correspondante et d'appliquer un algorithme de recherche de l'arbre couvrant de poids maximum sur la matrice ainsi créée. Le résultat est un arbre non orienté optimal maximisant l'information mutuelle par paire. D'autres mesures peuvent être utilisées pour la pondération, Heckerman et al. [1995] proposent la variation d'un score décomposable provoqué par l'ajout de chaque arête. L'orientation de l'arbre obtenu s'effectue par propagation à partir d'une variable racine (généralement choisie aléatoirement) vers les variables les plus éloignées de cette racine. Cette orientation ne créée donc aucune *v-structure* rendant tout arc inversible ce qui assure de conserver le même score indépendamment du choix de la variable racine. La principale conséquence est donc l'impossibilité de déduire une quelconque causalité dans les réseaux produits.

2.1.2.3 Espace de recherche des DAG

Lorsque l'espace de recherche n'est pas suffisamment contraint, les algorithmes d'apprentissage perdent leur caractère optimal et ils tentent seulement de déterminer un maximum local en espérant se rapprocher au mieux du maximum global. L'espace des graphes dirigés sans circuits (DAG) représente l'espace le plus naturel mais aussi l'espace le plus vaste. Les heuristiques présentées dans les sections suivantes sont toutes basées sur un principe de recherche locale. Démarrant d'une structure initiale, les heuristiques guident la recherche par modification locale de la structure en appliquant à chaque étape une ou plusieurs opérations dites élémentaires. Les opérations élémentaires sont au nombre de trois : l'ajout d'un arc, la suppression d'un arc et l'inversion d'un arc. L'inversion peut être vue comme une suppression immédiatement suivie d'un ajout, son utilité apparaît dans le cas d'heuristiques qui appliquent une seule opération à chaque étape et qui refusent que cette opération dégrade le score de la structure. Prenons l'exemple de l'inversion d'un arc permettant d'améliorer le score de la structure courante cependant l'opération de suppression seule dégrade le score et sera donc interdite

par l'heuristique rendant ainsi l'inversion en deux étapes impossible. On voit ici qu'une inversion en deux étapes est proscrite d'où la nécessité d'un opérateur spécifique réalisant l'opération en une seule étape. Campos et al. [2002] proposent par ailleurs d'enrichir cette opération d'inversion, en sélectionnant, après chaque inversion d'un arc entre deux variables, la configuration optimal des parents pour ces deux variables parmi l'union de leur parents courants.

Nous parlerons par la suite d'opérations positives lorsque celles-ci améliorent le score de la structure, négatives dans le cas d'une dégradation et nulles pour les opérations menant à deux structures de même score. Nous verrons que certaines méthodes appliquent plusieurs opérations élémentaires à chaque étape pour progresser plus efficacement dans l'espace de recherche. Nous définirons également en section 3.2 3.3 de nouveaux opérateurs dits 'complexes' permettant de sortir d'un minimum local à l'instar de l'opérateur d'inversion.

Approches déterministes Une méthode est dite 'déterministe' si et seulement si le résultat de l'exécution de la méthode ne varie pas pour un même ensemble de paramètres d'entrée. Ce comportement permet de prévoir l'intégralité de l'exécution facilitant l'analyse des résultats et l'identification des faiblesses de ces méthodes.

K2 Cooper and Hersovits [1992] décrivent l'algorithme $K2$ évoluant dans un espace des DAG contraint à un ordre donné. Sachant un ordre complet sur les variables, l'algorithme autorise uniquement pour chaque variable l'ajout de parents qui la précède dans cet ordre. Concrètement l'algorithme construit pour chaque variable l'ensemble de ses parents en ajoutant à chaque étape le parent autorisé au regard de l'ordre qui maximise le score sachant les parents précédemment ajoutés. La méthode telle que décrite par ses auteurs traite les variables suivant l'ordre fourni, cependant chacune des variables peut être traitée indépendamment des autres. En effet l'ordre imposé assure l'acyclicité des structures reconstruites or lorsque le score employé est décomposable seule la contrainte d'acyclicité empêche de traiter chaque variable indépendamment.
Les performances de cette méthode sont donc fortement liées au choix de cet ordre. Cependant le problème de recherche d'un ordre optimal est également NP-difficile. Cette recherche devient ainsi un problème en soit, une classe de méthodes que nous présentons en section 2.1.2.5 développe des heuristiques afin de parcourir l'espace des ordres où l'algorithme K2 est employé en tant que routine à chaque évaluation d'un ordre.
Il faut cependant noter que l'algorithme $K2$ n'est pas optimal bien que contraint dans son espace de recherche, l'ajout d'un seul parent à chaque étape de l'algorithme empêche la détection d'effets d'interaction de plusieurs variables qui, prisent indépendamment l'une de l'autre, n'améliorent pas le score. Ce problème est récurent pour les algorithmes qui n'appliquent qu'une seule opération élémentaire à chaque étape tout en refusant de dégrader le score. De plus l'algorithme K2 n'applique que des opérations d'ajout d'arc sans jamais considérer de suppression (les inversions sont également proscrites du fait de l'ordre) pourtant dans le cas d'effets d'interaction la présence d'un parent précédemment ajouté peut être remise en cause, après l'ajout d'un nouveau parent, dû aux changements induit sur les probabilités conditionnelles.

Greedy Search L'algorithme *greedy search* (noté GS), appelé également *hill-climbing*, progresse dans un espace non contraint. A chaque itération l'algorithme définit le voisinage de l'état courant du graphe. Chaque voisin correspond à une structure acyclique atteignable à partir de la structure courante lorsque l'on applique l'un des 3 opérateurs élémentaires, le calcul du score de chacune des structures est facilitée par l'utilisation d'un score décomposable. La structure voisine maximisant le score est alors sélectionnée, si celle-ci est meilleure que la structure courante alors l'opérateur associé est appliqué sinon la recherche s'arrête. L'algorithme converge alors sans garantie d'optimalité.
A l'instar de l'algorithme $K2$ dépendant de son ordre, GS requiert une structure initiale. Le graphe vide représente le choix le plus fréquent en l'absence de connaissances *a priori*. Une autre possibilité consiste à débuter d'une structure aléatoire, ce qui se révèle être en pratique un mauvais choix du fait des difficultés de l'algorithme à s'écarter de son point de départ. Nous expérimenterons en section 3.5.4.1 l'impact de l'initialisation sur l'apprentissage.
A chaque itération le voisinage de la structure courante constituée de p variables est de l'ordre de $O(p(p-1))$ (correspondant au nombre d'arcs possibles dans la structure). Afin de réduire cette complexité il est possible de restreindre l'espace de recherche. Dans cette optique Friedman et al. [1999b] au travers de leur algorithme *Sparse Candidate* sélectionnent pour chaque variable un nombre fixé de parents prometteurs à considérer lors de la recherche. Gámez et al. [2011] proposent quant à eux de restreindre progressivement le voisinage de chaque variable en interdisant pour toute la suite de la recherche une opération élémentaire dès lors que celle-ci vient à dégrader le score dans la configuration courante.

Tabu La recherche *Tabu* proposée par Glover and Laguna [1993] est une extension de l'algorithme GS. Alors que ce dernier s'arrête dès lors qu'aucun voisin n'est meilleur que le graphe courant, la recherche *Tabu* autorise de dégrader le score tout en sélectionnant à chaque itération la meilleure des structures voisines. L'algorithme retourne au terme de la recherche la meilleure structure rencontrée durant son exploration. L'objectif visé est de s'écarter suffisamment de l'optimum local atteint après avoir appliqué toutes les opérations positives en autorisant certaines opérations négatives menant par la suite à un meilleur optimum local. Pour cela il est nécessaire de définir une *liste Tabu* d'une taille k qui peut être fixe ou variable durant la recherche. Cette liste maintient en mémoire les k dernières opérations appliquées à la structure afin d'interdire toute remise en cause de ces opérations pour les k prochaines itérations. Sans la définition de cette liste, toute opération dégradant le score serait immédiatement suivie de l'opération inverse pour ramener le score à son niveau initial, empêchant ainsi la recherche d'explorer d'autres structures.
La taille de la liste est donc un paramètre important, une liste trop courte empêche de s'écarter suffisamment de l'optimum local tandis qu'une liste sur-dimensionnée empêcherait d'atteindre un meilleur optimum. Une solution alternative consiste à utiliser une liste de taille aléatoire, redéfinie après chaque itération, variant entre deux bornes fixées. L'arrêt de la recherche peut être fixée suivant un ou plusieurs critères tel qu'une limite sur le nombre total d'itérations effectuées ou sur le nombre d'itérations effectuées depuis la découverte du dernier optimum local.

LAGD Holland et al. [2008] proposent l'algorithme LAGD basé sur l'algorithme GS mais qui explore un voisinage plus large en appliquant à chaque itération k opérations élémentaires avec $k > 1$ ($k = 1$ est équivalent à GS). Le nombre de structures voisines atteignables après k opérations croit rapidement ($\prod_{i=0}^{k} p - i$) ce qui empêche une exploration exhaustive de ce voisinage. Pour cette raison un second paramètre de l'algorithme permet de ne conserver que les l meilleures opérations au niveau i pour lesquelles l'application d'une seconde opération au niveau $i + 1$ sera considérée, réduisant ainsi la taille du voisinage à $p\ (p - 1)\ l^{k-1}$.
L'utilisation de cette méthode requiert un compromis dans le réglage de ces deux paramètres, une exploration basée sur l'application d'un grand nombre d'opérations à chaque étape devra restreindre les directions dans lesquelles cette exploration s'effectue. Et à l'inverse une recherche avec une exploration quasi-exhaustive du voisinage sera limitée dans la profondeur de son déplacement à chaque étape.

Optimal reinsertion Une autre approche visant à appliquer plusieurs opérations à chaque étape est celle proposée par Moore and Wong [2003]. Leur méthode nécessite une structure initiale de bonne qualité, typiquement issue d'une recherche de GS, puis le voisinage (dans le graphe) de chaque variable est successivement remis en cause. A chaque étape une variable cible est sélectionnée dont toutes les arcs entrants ou sortants sont supprimés. Puis l'algorithme détermine un nouvel ensemble de parents et d'enfants optimal pour cette variable sous contrainte d'un nombre de parents limité. Un algorithme de *Branch and Bound* permet de limiter le nombre de configurations à explorer accélérant ainsi l'algorithme. Ce processus est répété pour chaque variable jusqu'à ce que plus aucune modification ne soit nécessaire.
Cette méthode permet de détecter systématiquement les situations d'effet d'interaction de deux parents sur une même variable à la différence des méthodes présentées précédemment.

Approches stochastiques A la différence des approches déterministes, les méthodes stochastiques fournissent un résultat pouvant varier pour un même ensemble de paramètres d'entrée. Ces méthodes permettent si elles sont exécutées suffisamment longtemps ou répétées un certains nombre de fois d'obtenir des résultats de bonnes qualités. Cependant le réglage de leurs paramètres est bien souvent complexe.

Algorithme de Metropolis-Hastings Nous pouvons utiliser le principe des chaînes de Markov cachées pour l'apprentissage de la structure d'un RB, dans ce cas à chaque état de cette chaîne correspond une structure du RB et chaque transition à l'application d'une opération élémentaire [Madigan and York, 1995]. L'algorithme se déroule sous la forme d'une marche aléatoire dans cette chaîne régit par une probabilité d'acceptance de type *Metropolis-Hastings* telle que la probabilité de transition d'un graphe initial $\mathcal{G}$ vers un des graphes acycliques voisins $\mathcal{G}'$ est

$$\mathbb{P}(\mathcal{G} \rightarrow \mathcal{G}') = min(1, \frac{\mathbb{P}(\mathcal{G}', D)\ T(\mathcal{G}' \rightarrow \mathcal{G})}{\mathbb{P}(\mathcal{G}, D)\ T(\mathcal{G} \rightarrow \mathcal{G}')})$$

où $T(\mathcal{G} \rightarrow \mathcal{G}')$ représente la probabilité *a priori* d'effectuer la transition de $\mathcal{G}$ vers $\mathcal{G}'$, en l'absence de connaissance experte cette probabilité est uniforme pour toutes les transitions vers un graphe $\mathcal{G}'$.

Dans le cas où le score utilisé s'écrit comme le logarithme de la probabilité jointe $\mathbb{P}(\mathcal{G}', D)$, on peut définir le comportement de $\mathbb{P}(\mathcal{G} \rightarrow \mathcal{G}')$ ainsi

$$\begin{cases} si \quad Score(\mathcal{G}'|D) > Score(\mathcal{G}|D) \quad \mathbb{P}(\mathcal{G} \rightarrow \mathcal{G}') = 1 \\ sinon \quad \mathbb{P}(\mathcal{G} \rightarrow \mathcal{G}') = \exp(Score(\mathcal{G}'|D) - Score(\mathcal{G}|D)) \end{cases}$$

Une opération positive est donc appliquée avec une probabilité de 1. Dans le cas d'une opération négative, la probabilité d'être appliquée décroit lorsque la dégradation du score augmente, limitant ainsi l'application d'opérations très coûteuses en terme de score. L'unique paramètre de cet algorithme indique le nombre d'itérations à effectuer avant que la meilleure structure rencontrée ne soit retournée. Il est également possible de mémoriser régulièrement durant la recherche l'état de la structure afin de fournir à terme une fréquence d'apparition des arcs dans les structures retenues plutôt qu'un unique réseau final, dans ce cas il convient de fixer la fréquence de ces observations.

Simulated Annealing Janžura and Nielsen [2006] utilisent l'algorithme de recuit simulé qui est une évolution de l'algorithme MCMC où un paramètre t représentant une *température* permet de faire tendre la probabilité d'effectuer une opération vers un choix déterministe.
On peut écrire cette probabilité en fonction de la température t

$$\begin{cases} si \quad Score(\mathcal{G}'|D) > Score(\mathcal{G}|D) \quad \mathbb{P}(\mathcal{G} \rightarrow \mathcal{G}') = 1 \\ sinon \quad \mathbb{P}(\mathcal{G} \rightarrow \mathcal{G}') = \exp(\frac{Score(\mathcal{G}'|D) - Score(\mathcal{G}|D)}{t}) \end{cases}$$

Concrètement une température t_{init} doit être fixée ainsi qu'une règle de décroissance de cette température du type *faire diminuer t d'une valeur* δ_t *toutes les x itérations*. Lorsque t est élevée la probabilité d'accepter une opération négative est forte même si celle-ci dégrade nettement le score de la structure, puis progressivement la diminution de t entraîne une stringence plus forte de l'acceptance jusqu'à $t \approx 0$ où la probabilité d'accepter une opération négative est nulle. L'algorithme se termine alors par une phase de type GS et retourne la meilleure structure rencontrée durant la recherche. Il est possible de fixer un système de réchauffement où t augmente à nouveau permettant d'explorer d'autres configurations, le nombre de ces cycles de chauffe-refroidissement constitue un nouveau paramètre.

Un système de recuit simulé a également été utilisé par Elidan et al. [2002] dans le but cette fois de perturber les données lors d'une recherche de type GS.

Algorithmes évolutionnaires Si les réseaux bayésiens peuvent servir à modéliser des phénomènes biologiques, il est possible d'effectuer la démarche inverse et de se servir de concepts du monde du vivant afin d'apprendre la structure d'un RB. Les approches évolutionnaires en font partie, ces approches utilisent le concept de *population* afin de représenter un ensemble de DAG. Cette population est générée initialement de façon aléatoire puis au travers d'opérateurs de croisement et de mutation, cette population évolue de façon stochastique guidée par la fonction de score. Après un nombre fixe de générations, l'algorithme s'arrête rendant la meilleure structure rencontrée durant la recherche.
Parmi ces approches citons les algorithmes génétiques [Larrañaga et al., 1996], la programmation évolutionnaire [Wong et al., 1999] ou des méthodes basées sur les EDA (Estimation of Distribution

Algorithms) où l'évolution de la population suit une distribution de probabilité estimée sur un sous ensemble de la population [Blanco et al., 2003, Thibault et al., 2008]. Plus récemment des schémas d'évolution plus complexes tels que l'évolution coopérative [Barrière et al., 2009] ont été proposés.

Méthodes ensemblistes Nous avons vu jusqu'à maintenant des méthodes permettant de sélectionner une unique structure maximisant localement le score considéré. Cependant nous pouvons nous interroger sur la raison de ne sélectionner que la meilleure structure tout en ignorant les nombreuses autres structures ayant des scores proches de l'optimum local. L'objectif des approches ensemblistes est de produire un ensemble de structures candidates d'où peut être déduit une structure consensus en conservant les arcs les plus fréquemment rencontrés à l'aide d'un seuil. L'acyclicité de cette structure peut être restaurée si besoin en supprimant un ensemble d'arcs parmi les moins fréquents, la recherche de cet ensemble optimal est cependant un problème NP-dur [Karp, 1972]. Le DAG obtenu ne présente alors aucune propriété d'optimalité du score mais assure la robustesse de la structure. Pour cette raison les méthodes à base d'ensemble ne retournent généralement que la fréquence d'apparition de chacun des arcs indiquant ainsi leur robustesse et non un DAG. Des travaux plus récents proposent de fusionner un ensemble de DAG au travers des graphes essentiels correspondants afin de distinguer par la suite les arêtes fréquemment apprises pour lesquelles l'une des deux orientations prédomine [Nguyen, 2012]. Nous présentons ci-dessous 3 techniques permettant d'obtenir un ensemble de structures candidates.

Boostraping de données Le *bootstrap* avec remise (appelé parfois *bagging*) permet de générer à partir d'un ensemble D de N observations, une collection d'ensembles composés pour chacun de N tirages avec remise parmi D. Le tirage s'effectue de façon équiprobable pour chaque observation. Nous obtenons ainsi différents jeux de données sur lesquels nous pouvons appliquer les méthodes classiques d'apprentissage et obtenir différents DAG candidats [Friedman et al., 1999a].

Boosting de méthodes Issu du monde de la classification, le *boosting* combine les résultats de différents classifieurs afin d'en améliorer les performances. Nous pouvons appliquer ce principe dans le cadre de l'apprentissage de structure en combinant les résultats de différents algorithmes. Un exemple d'utilisation pour l'analyse de données biologiques sera présenté dans la section 7.2.2 sous le terme de *méta-analyse*. Cette technique comme celle du boostrap, peut employer toute méthode d'apprentissage indépendamment des modèles statistiques utilisés.

MCMC Certaines méthodes, notamment les approches stochastiques sont parfaitement adaptées à une approche ensembliste. La méthode des MCMC par son principe de marche aléatoire permet de rencontrer à plusieurs reprises durant la recherche des structures de bonne qualité. Il est donc possible de retenir régulièrement l'état de la structure courante afin d'obtenir un ensemble de DAG.

2.1.2.4 Espace de recherche des cpDAG

Lors de l'utilisation d'un score équivalent, qui associe un même score aux structures ayant la même probabilité jointe, les méthodes précédentes doivent effectuer un choix d'orientation lors de

l'ajout de certains arcs menant à deux structures équivalentes. Ce choix dépend généralement de l'ordre de définition des variables dans le programme (lexicographique par exemple) ce choix n'est cependant pas anodin puisqu'il conditionne la suite de la recherche. La recherche dans l'espace des cpDAG permet de palier cette difficulté en définissant le voisinage comme étant l'ensemble des classes d'équivalence atteignables à partir des différentes instanciations de la classe d'équivalence courante, en y appliquant un des opérateurs classiques [Auvray and Wehenkel, 2002]. Cette recherche équivaut à explorer le voisinage de chaque instance (DAG) possible du cpDAG courant, puis à factoriser ces DAG par l'ensemble des cpDAG associés. Cependant en pratique le nombre exponentiel d'instanciations possibles d'un cpDAG empêche d'utiliser une projection du problème dans l'espace des DAG et nécessite de raisonner uniquement dans l'espace des cpDAG.

Bien que cet espace soit plus restreint que l'espace des DAG, cette réduction n'est en réalité que d'un facteur inférieur à 4 [Gillispie and Perlman, 2001]. Une recherche exhaustive reste donc toujours impossible, nous décrivons ci-dessous deux algorithmes permettant de parcourir cet espace de recherche.

GES L'algorithme GES introduit par Chickering and Maxwell [2002] se compose de deux phases distinctes. Débutant d'une structure vide, la première phase consiste à explorer l'espace des cpDAG en considérant uniquement les opérations d'ajout d'un arc. L'ajout qui maximise le score employé est alors appliqué au cpDAG courant, si aucune *v-structure* ne l'impose l'arc correspondant n'est pas orienté dans le graphe. L'exploration se poursuit de cette manière tant que des opérations d'ajout positives existent. Dans le cas où les observations sont représentables par un DAG de loi jointe $\mathbb{P}_D$, à l'issu de cette première phase, toutes les dépendances décrites par $\mathbb{P}_D$ sont également présentes dans le cpDAG courant G. La réciproque n'est cependant pas exacte, certaines indépendances de $\mathbb{P}_D$ ne sont pas représentées dans G, la deuxième phase de l'algorithme vise donc à supprimer certains arcs afin d'obtenir un cpDAG représentant la *carte d'indépendance parfaite* de $\mathbb{P}_D$. Cet algorithme est optimal dans le cas de données représentable par une loi jointe qui est elle même représentable par un DAG et d'un nombre d'observation suffisamment grand pour détecter sans erreur les indépendances de $\mathbb{P}_D$. Deux conditions rarement valides dans le cas de données réelles.

KES Nielsen et al. [2003] observent que le caractère glouton de l'algorithme GES ne permet pas d'apprendre la structure optimale pour certaines lois jointes. L'algorithme KES représente une variante de l'algorithme GES, où l'opération appliquée correspond à l'opération optimale parmi un sous-ensemble aléatoire des opérations positives. La taille de ce sous-ensemble est définie suivant un paramètre k réglable entre 0 et 1. Lorsque $k = 1$ le sous ensemble contient toutes les opérations positives ce qui équivaut à GES, à l'inverse lorsque k est proche de 0 la taille réduite du sous-ensemble augmente le caractère aléatoire de KES. Contrairement à GES, KES nécessite plusieurs exécutions avant d'atteindre une solution optimale (pour $k \neq 1$). Les conditions d'adéquation des données à la loi jointe et d'un grand nombre d'observations, sont toujours nécessaires afin d'assurer le caractère optimal de l'algorithme.

Espace des rpDAG Une alternative à mi chemin entre les DAG et les cpDAG est proposée par Acid and de Campos [2003] sous le nom de rpDAG (*restricted acyclic partially directed graphs*). Alors qu'une classe d'équivalence est représentée par un seul et unique cpDAG (voir section 1.3), plusieurs rpDAG peuvent représenter la même classe d'équivalence. Le passage d'un cpDAG vers son ou ses rpDAG associés se fait par l'orientation des cycles contenus dans le cpDAG, on peut ainsi voir un rpDAG comme une version pseudo-orientée du cpDAG correspondant. Cette représentation permet à la fois de déterminer rapidement les instanciations possibles d'un rpDAG grâce aux orientations supplémentaires mais également de déterminer plus facilement si un pDAG est effectivement un rpDAG. Il s'agit ainsi de faire un compromis entre le fait de devoir sélectionner un rpDAG qui ne représente pas à lui seul l'intégralité d'une classe d'équivalence mais de permettre en contre partie une exploration du voisinage facilitée par l'identification aisée des rpDAG.

2.1.2.5 Espace de recherche des ordres

Le problème majeur de la recherche dans l'espace des DAG mais aussi des pDAG est le caractère local des modifications induites sur la structure à chaque itération. Malgré les améliorations développées dans les sections précédentes il est parfois difficile de s'écarter d'un optimum local par le biais de modifications structurelles minimes. Il est alors possible d'utiliser un autre espace de recherche, notamment celui des ordres sur les variables tel qu'il a été introduit dans le contexte de l'algorithme K2. Modifier l'ordre des variables induit des changements importants sur la structure apprise ce qui permet de parcourir plus largement l'espace des DAG. De plus la recherche d'une structure de bonne qualité sachant un ordre est relativement peu coûteuse et peut être utilisée comme une routine d'un algorithme parcourant l'espace des ordres possibles.

Afin de parcourir cet espace il est nécessaire de spécifier 3 éléments :

- l'heuristique de parcours dans l'espace des ordres
- les opérations de modification de l'ordre courant afin de définir les ordres voisins
- la stratégie pour apprendre la meilleure structure compatible à un ordre donné

La recherche de la meilleure structure compatible à un ordre donné peut s'effectuer par énumération de toute les structures compatibles. Dans ce cas il est nécessaire de fixer des limites sur le nombre de parents maximum ainsi que sur les parents potentiels pour chaque nœuds avant la recherche. Parmi les méthodes utilisant cette technique nous pouvons citer Friedman and Koller [2003] qui emploient un algorithme MCMC afin de parcourir l'espace des ordres. Les modifications sur l'ordre sont obtenues par permutation de deux variables ou par scission de l'ordre en deux puis par permutation de ces deux sous-ensembles. Teyssier and Koller [2005] utilisent quant à eux un algorithme GS muni d'une liste Tabu afin de se déplacer dans l'espace des ordres en ne considérant que les permutations de variables successives dans l'ordre courant.

Une autre méthode permettant de déterminer la meilleur structure compatible à un ordre, consiste à utiliser l'algorithme K2. Alonso-Barba et al. [2011] proposent d'utiliser une version modifiée de l'algorithme K2 dans ce but (enrichi de l'opération de suppression nommée alors $K2M$), puis

d'employer un algorithme GS muni d'un opérateur d'insertion afin de parcourir l'espace des ordres. L'opérateur d'insertion utilisé consiste à modifier la position d'une seule variable dans l'ordre courant.

2.1.3 Approches hybrides

Nous avons vu jusqu'à maintenant les méthodes basées sur des tests d'indépendances puis celles utilisant une fonction de score. Il est cependant possible d'utiliser des éléments des deux approches au sein d'une même méthode dite *hybride* afin de tirer partie des avantages de chacune d'elles. Ce rapprochement peut s'effectuer sous plusieurs formes que nous détaillons dans les sections suivantes. La première possibilité est d'utiliser un score défini à partir des mesures d'indépendance tel l'algorithme BENEDICT [Acid and de Campos, 2001]. Une autre approche consiste à alterner successivement une approche à base de score et une recherche d'indépendances, enfin nous pouvons utiliser la recherche d'indépendances comme un moyen de réduire l'espace de recherche *a priori* sur lequel s'effectue une recherche à base de score.

Algorithme BENEDICT L'algorithme BENEDICT [Acid and de Campos, 2001] s'apparente en plusieurs points à l'algorithme K2, la recherche s'effectue de manière gloutonne par ajout successif de l'arc qui maximise un score en restant compatible avec un ordre fixé initialement. Le caractère hybride de l'algorithme provient du score qui est défini comme étant la somme des divergences de *Kullback-Leibler* de chaque indépendance conditionnelle induite par le graphe sachant les données. Seules les indépendances conditionnelles dont l'ensemble de conditionnement est minimal sont évaluées afin de limiter la complexité de la méthode. Ce score n'est toutefois pas décomposable car l'ajout d'un seul arc peut modifier les indépendances liées à toutes les variables du réseau. De plus la divergence de *Kullback-Leibler* ne pouvant être négative, il est nécessaire de fixer un critère d'arrêt spécifique. Une des possibilités consiste à réduire durant la recherche l'ensemble des arcs potentiels $\mathcal{L}$. Après chaque ajout, les arcs reliant deux variables testées indépendantes conditionnellement au graphe courant sont retirés de $\mathcal{L}$. La recherche s'arrête lorsque $\mathcal{L} = \emptyset$. Une dernière phase consiste alors à retirer certains arcs du graphe final *a posteriori* en testant chacun d'eux selon leur ordre d'insertion dans le graphe.

Approches hybrides séquentielles Les approches hybrides que nous appelons *séquentielles* alternent successivement au cours de leur exécution, une ou plusieurs phases d'optimisation d'un score et de recherche d'indépendances. Ces méthodes offrent l'avantage d'être modulaires, chaque phase étant indépendante il est possible d'imaginer différentes combinaisons.

Citons par exemple De Campos et al. [2003] qui proposent dans leur algorithme *Iterated Hill Climbing* d'utiliser le test d'indépendance du χ^2 afin de corriger les structures fournies par l'algorithme *Greedy Search* (GS). Après une première exécution de GS, chaque couple de variables est testé conditionnellement au graphe courant. Si deux variables reliées dans le graphe sont indépendantes d'après le test du χ^2 alors l'arc correspondant est supprimé tandis qu'un arc est ajouté lorsqu'il apparaît une dépendance qui n'est pas présente dans le graphe. Ce nouveau graphe sert alors de point de départ à une nouvelle exécution de GS. Ce processus est répété un nombre fixé de fois et permet de palier

aux erreurs commises par GS durant la phase de reconstruction grâce à des tests d'indépendance *a posteriori*.

Restriction de l'espace de recherche Les tests d'indépendance peuvent être également utilisés afin de restreindre l'espace de recherche avant même l'exécution d'un algorithme à base de score. Cette approche se révèle particulièrement intéressante dans le cas de grands graphes afin de réduire *a priori* le nombre de DAG possibles.

Wong and Leung [2004] utilisent par exemple le test du χ^2 dans une recherche basée sur la programmation évolutionnaire. Dans un premier temps les tests d'indépendance du χ^2 d'ordre 0 et d'ordre 1 sont effectués pour chacun des arcs possibles puis la plus haute *p-value* est conservée pour chacune d'elles. La population initiale est alors constituée d'un ensemble de graphes possédant chacun une valeur aléatoire de risque α, la structure de chaque graphe est construite par tirages aléatoires dans l'ensemble des arcs dont la *p-value* est supérieure au risque $1 - \alpha$ associé au graphe. Un algorithme de programmation évolutionnaire classique est ensuite appliqué pour un nombre de générations fixé. Chaque graphe généré durant le processus hérite d'un risque α proche de celui de son parent.

Recherche de la couverture de Markov La notion de couverture de Markov introduite par Pearl [1988] peut également être utilisée dans le but de limiter l'espace de recherche *a priori*.

Definition 2.1.1: *Soit $\mathcal{MB}_X$ un ensemble de variables et X une variable appartenant au graphe $\mathcal{G}$ telle que $X \notin \mathcal{MB}_X$. $\mathcal{MB}_X$ est une couverture de Markov de X dans $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ si et seulement si $X \perp \mathcal{X} \setminus (\mathcal{MB}_X \cup \{X\}) | \mathcal{MB}_X$ et $\mathcal{MB}_X$ est minimal.*

Graphiquement la couverture de Markov $\mathcal{MB}_X$ d'une variable X est composée de ses parents, de ses enfants et des parents des enfants de X. Il est donc aisé de définir $\mathcal{MB}_X$ graphiquement lorsque le graphe entier est connu, cependant la recherche du graphe entier représente une tâche plus difficile que ne peut l'être la recherche de la couverture de Markov seule pour chaque variable. Koller and Sahami [1996] introduisent alors l'idée de développer des algorithmes spécifiques pour déterminer la couverture de Markov des différentes variables. Il est alors possible d'utiliser la composition de ces couvertures de Markov lors d'une recherche classique à base de score en autorisant uniquement les arcs entre une variable et un des membres de sa couverture de Markov.

Parmi les différentes approches employées afin de définir les couvertures de Markov citons Koller and Sahami [1996] qui proposent d'utiliser les corrélations de Pearson de degré 0 ou Aliferis et al. [2002] qui utilisent l'information mutuelle conditionnelle dans leur algorithme IAMB ainsi que dans ses nombreuses déclinaisons [Tsamardinos et al., 2003a]. L'algorithme HITON défini par Aliferis et al. [2003a] utilise aussi bien l'information mutuelle, le rapport de vraisemblance ou encore le test de Fisher ce qui démontre la possibilité d'utiliser divers tests d'indépendance pour ces algorithmes.

La recherche de la couverture de Markov pour une variable cible X s'effectue généralement en deux étapes de type *forward-backward*. Nous illustrons ce processus sur la Figure **2.1**, dans cet exemple

nous cherchons à déterminer la couverture de Markov de la variable X_4. La phase *forward* consiste à inclure progressivement dans $\mathcal{MB}_{X_4}$ toutes les variables dépendantes de X_4 sachant les variables déjà présentes dans $\mathcal{MB}_{X_4}$. Nous obtenons à ce stade l'ensemble des variables, rangées suivant leur ordre d'insertion dans $\mathcal{MB}_{X_4}$, $\mathcal{MB}_{X_4} = \{X_6, X_1, X_2, X_3\}$ (Fig. **2.1**(b)), notons que X_3 est dépendante de X_4 dès lors que $X_6 \in \mathcal{MB}_{X_4}$, à l'inverse les variables X_5 et X_7 sont indépendantes de X_4 dès que $X_2 \in \mathcal{MB}_{X_4}$. La phase *backward* reconsidère chaque variable appartenant à $\mathcal{MB}_{X_4}$ afin de vérifier si la dépendance conditionnelle est maintenue sachant tout sous ensemble de $\mathcal{MB}_{X_4}$ possible. Cette étape provoque le retrait de X_3 de $\mathcal{MB}_{X_4}$ (Fig. **2.1**(c)) dès lors que l'ensemble de conditionnement n'inclue pas X_6.

Ces deux étapes ne permettent pas d'identifier la couverture de Markov complète puisque les parents des enfants de X_4 (ie les époux/ses de X_4) sont ôtés de $\mathcal{MB}_{X_4}$ lors de la phase *backward*. Seul l'ensemble des parents et des enfants directs de X_4 (noté $\mathcal{PC}_{X_4}$) est alors détecté par ces deux phases. Afin d'obtenir $\mathcal{MB}_{X_4}$ il faut non seulement déterminer $\mathcal{PC}_{X_4}$ mais aussi les ensembles $\mathcal{PC}_Y$ tels que $Y \in \mathcal{PC}_{X_4}$, ce qui correspond à déterminer les parents et les enfants directs pour chacun des parents et des enfants directs de X_4. L'union des ensembles $\mathcal{PC}_Y$ inclue par conséquent les époux de X_4 mais aussi d'autres variables telles que X_5 qui appartient à PC_{X_2} (Fig. **2.1**(d)). Afin d'extraire uniquement les époux/ses de X_4 il faut tester cette fois pour chaque variable appartenant à $\mathcal{PC}_Y$ si il existe une variable appartenant à $\mathcal{PC}_{X_4}$ qui permet, si elle est ajoutée à l'ensemble de conditionnement, de casser l'indépendance observée lors des deux phases précédentes. Dans le cas de X_3 il suffit d'ajouter X_6 à l'ensemble de conditionnement pour retrouver la dépendance vis-à-vis de X_4. Concernant X_5 testée indépendante de X_4 lors de la phase *forward* sachant X_2 il n'existe aucune variable à ajouter permettant de trouver une dépendance dès lors que la présence de X_2 bloque toute dépendance (Fig. **2.1**(e)).

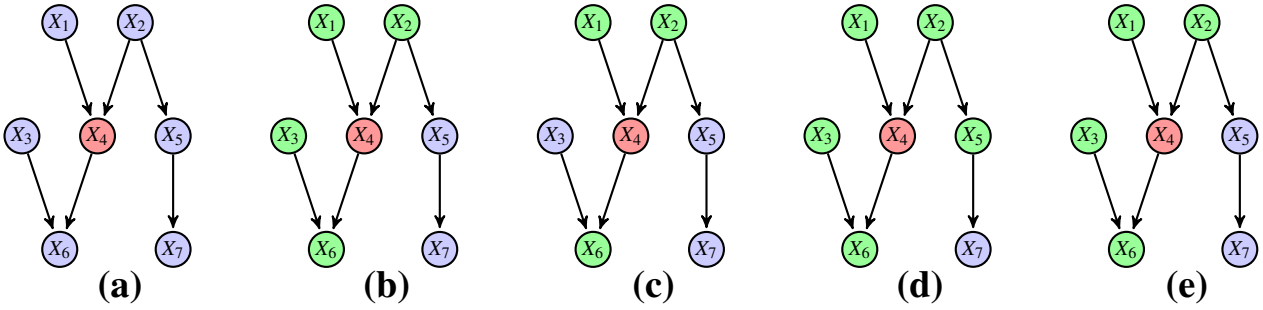


Figure 2.1 – Recherche de la couverture de Markov de X_4 indiquée en rouge. Les arcs représentent la vraie structure. Les nœuds en verts correspondent aux variables appartenant à $\mathcal{MB}_{X_4}$ en cours de construction après chacune des étapes : (a) état initial, (b) étape forward, (c) étape backward, (d) extension aux couvertures de Markov voisines (e) état final

Ces différentes étapes sont clairement définies dans l'algorithme MMMB de Tsamardinos et al. [2003b] dont la recherche des ensembles $\mathcal{PC}$ est effectuée par l'algorithme MMPC.

Aliferis et al. [2010] effectuent une comparaison récente des différentes méthodes d'apprentissage des couvertures de Markov montrant la supériorité de l'algorithme HITON muni d'un filtre statistique afin de restreindre *a priori* les variables candidates.

Plutôt que de restreindre la recherche aux arcs reliant une variable à sa couverture de Markov, il est donc possible de s'arrêter aux seuls ensembles retournés par la recherche *forward-backward* permettant d'identifier les parents et enfants directs. L'espace de recherche est alors sensiblement réduit. Cette technique est utilisée par Tsamardinos et al. [2006] avec leur algorithme MMHC dont le but est cette fois de reconstruire le réseau complet. Dans un premier temps l'ensemble des parents et enfants directs est identifié pour chaque variable par l'algorithme MMPC puis une recherche de type GS est effectuée à partir du graphe vide en autorisant uniquement les arcs $X \rightarrow Y$ avec $X \in \mathcal{PC}_Y$.

Filtre De manière similaire à la recherche de couvertures de Markov, l'utilisation d'un filtre permet de sélectionner pour chaque variable un ensemble de parents potentiels. La recherche n'explorera alors que des graphes composés uniquement d'arcs autorisés. L'algorithme *Sparse Candidate* de Friedman et al. [1999b] illustre parfaitement cette approche. L'algorithme itère une succession de deux étapes. Dans un premier temps pour chaque variable sont conservés les k meilleurs parents au sens de l'information mutuelle ou d'une fonction de score conditionnellement à la structure actuelle. La seconde étape est composée d'une recherche via l'algorithme GS amélioré par une liste $TABU$. Ces deux étapes sont répétées tant que le score de la structure n'évolue plus par rapport à l'itération précédente.

Afin d'adapter la sélection des parents candidats à chaque variable sans fixer un nombre commun de parents potentiels Schmidt et al. [2007] proposent d'utiliser la régression linéaire pénalisée de type $L1$ comme filtre. Pour chaque variable une valeur spécifique du paramètre λ est déterminée par maximisation du score MDL, puis une régression sur cette variable permet de définir la liste de ses parents potentiels. La taille de ces listes varie pour chaque variable en fonction de la valeur du paramètre λ sélectionnée.

Une autre approche proposée par Goldenberg and Moore [2004] utilise le principe d'ensemble de co-occurence pour des variables binaires (état 0/1). Ces ensembles regroupent des variables présentes à l'état 1 pour les mêmes observations. A partir de ces ensemble une liste d'arcs potentiels est définie, sur laquelle est appliquée une recherche de type GS. Cependant afin de limiter la taille des ensembles de co-occurence, cette approche est limitée au cas de données *sparses*, pour lesquelles une majorité des observations correspond à l'état 0.

Le principal risque dans cette restriction de l'espace de recherche vient du caractère définitif de la sélection. Une mauvaise sélection induit obligatoirement une structure apprise de piètre qualité. De manière générale ces restrictions doivent être effectuées avec parcimonie, car si un faux arc autorisé n'est pas obligatoirement ajouté durant la recherche, un vrai arc interdit par le filtre ne le sera jamais. Nous proposons en section 4.2 d'utiliser un filtre basé sur la même fonction de score que celle employée lors de la recherche de la structure de la même manière que Friedman et al. [1999b].

2.2 Discrétisation des données

Dans cette thèse nous nous sommes placés dans le formalisme des réseaux bayésiens discrets, ce qui impose naturellement une phase de discrétisation lors du traitement de données continues telles que peuvent l'être certaines données biologiques. Cette discrétisation étant la première des étapes menant à l'apprentissage de la structure il revient de ne pas la négliger. En effet une perte d'information trop importante lors d'une discrétisation hasardeuse hypothèque par la suite toute possibilité d'apprendre un réseau de bonne qualité. Le choix d'une méthode de discrétisation efficace, c'est-à-dire qui limite la perte d'information, est donc un problème en soit.

2.2.1 Méthodes de discrétisations

Considérons N observations d'une variable continue X, discrétiser X en K classes revient à associer à chaque valeur réelles $x_i \in X$ une valeur entière $x_i^K \in \{0, .., K-1\}$, nous obtenons ainsi le vecteur discrétisé X^K. L'ensemble des valeurs discrètes possibles est généralement constitué des K premières valeurs entières $\{1, .., K\}$, cependant cet ensemble peut également être constitué de valeurs non successives, négatives ou même qualitatives.
Ces associations valeurs réelles/discrètes s'effectuent suivant une *politique de discrétisation*. Une *politique de discrétisation* en K classes peut être définit sous la forme d'un vecteur Δ composé de $K+1$ valeurs réelles tel que :

- $\Delta_0 = -\infty$
- $\Delta_K = +\infty$
- $\Delta_i < \Delta_{i+1}$ pour tout $i \in \{0, .., K-1\}$

Le vecteur Δ définit les bornes de chacune des K classes et permet donc d'associer à chaque valeur réelle de X une valeur discrète tel que

$$\Delta_j \leq x_i < \Delta_{j+1} \Leftrightarrow x_i^K = j \quad \forall i \in \{0, .., N-1\} \; \forall j \in \{0, .., K-1\}$$

Toute méthode de discrétisation cherche donc à définir une telle politique, afin de classer ces méthodes, nous pouvons utiliser deux caractéristiques majeures. La première utilise le fait que la politique de discrétisation soit guidée ou non par la valeur des observations, nous parlons alors respectivement d'une discrétisation *supervisée* ou *non-supervisée*. La deuxième caractéristique indique si chaque variable est discrétisée de façon indépendante par rapport aux autres, dans ce cas nous parlons d'une discrétisation *univariée*, ou à l'inverse *multivariée* lorsque la discrétisation tient compte des interactions entre variables. Nous présentons par la suite 3 approches univariées à savoir la discrétisation par intervalle et par quantile qui sont des approches non-supervisées, ainsi que la recherche de modèle de mélange qui est supervisée. Nous décrirons également une approche multivariée et supervisée, basée sur la conservation de l'information mutuelle par paire.

2.2.1.1 Méthodes univariées

Par intervalle La discrétisation par intervalle utilise une politique basée sur une répartition uniformément espacée des bornes le long du domaine de définition des variables continues. Ainsi

pour une variable continue X prenant ses valeurs dans $[X_{min}, X_{max}]$, la politique de discrétisation en K classes se définit comme suit

$$\Delta = \{-\infty, X_{min} + \frac{(X_{max} - X_{min})}{K}, X_{min} + 2\ \frac{(X_{max} - X_{min})}{K},\ \ ..\ \ , +\infty\}$$

Cette discrétisation rapide à mettre en œuvre présente l'inconvénient majeur de ne tenir aucun compte de la répartition des observations de X le long de son domaine de définition. Ainsi si les observations divergent fortement, cette politique peut mener à la création de classes vides qui auraient pu, dans le cas d'un nombre de classes fixé, permettre de re-discrétiser des classes non vides.

Par quantile La discrétisation par quantile s'attache cette fois à une répartition uniforme des observation dans chacune des classes créées. Ainsi pour une variable continue X, la politique de discrétisation en K classes des N observations classées par ordre croissant de leur valeur se définit comme suit

$$\Delta = \{-\infty, x_{\frac{N}{K}}, x_{2\ \frac{N}{K}},\ \ ..\ \ , +\infty\}$$

où $x_{\frac{N}{K}}$ représente la $\frac{N}{K}$ème observation triée pour la variable X.

Une variante de cette approche souvent utilisée dans le cas de données biologiques, consiste à établir une discrétisation par *quantiles extrêmes*. Généralement une discrétisation en 3 classes est attendue permettant de classer chaque observation comme étant un état *sous-exprimées*, *normale* ou *sur-exprimées* de la variable. Les quantiles peuvent alors être fixés pour obtenir une répartition de 5%/90%/5% ou encore 10%/80%/10% de la population dans les 3 classes, l'idée étant de garder un caractère rare à la sur ou sous expression d'une variable.
La discrétisation par quantile permet à la différence de la discrétisation par intervalle de tenir compte de la répartition des observations et ne plus créer de classes sans observations. Cependant le fait d'imposer au préalable le pourcentage d'observations dans chaque classe peut amener à rassembler au sein d'une même classe des observations hétérogènes, voir même séparer dans deux classes différentes des observations ayant la même valeur et donc perdre de l'information.

Modèle de mélange Le problème de la classification automatique est un champ d'étude touchant à la fois le domaine de l'intelligence artificielle et des statistiques dont les objectifs peuvent êtres assimilés au problème de la discrétisation. Classifier un objet, une valeur, revient à lui attribuer une classe ce qui correspond exactement à l'objectif d'une discrétisation où les classes correspondent à des valeurs discrètes. Il est donc possible d'utiliser des techniques issus du monde de la classification afin de discrétiser des observations.

K-means L'algorithme des K-means cherche à regrouper les observations en K classes. Une observation appartient à la classe dont la distance entre la valeur de l'observation et la moyenne des valeurs des observations appartenant à cette classe est minimum. L'algorithme débute par une initialisation des moyennes des K classes (généralement aléatoire) puis agit par itération successive. Chaque itération débute par la classification de toutes les observations suivant la distance minimum

aux valeurs courantes des K moyennes, puis les moyennes des K classes sont recalculées suivant cette classification ce qui entraîne une nouvelle itération. L'algorithme converge lorsque aucun changement de classe n'est opéré entre deux itérations successives. Le résultat de la classification dépend donc de l'initialisation des K-moyennes, une solution consiste à exécuter plusieurs fois l'algorithme en faisant varier l'initialisation puis à conserver la meilleure classification au sens du minimum des distances euclidiennes entre chaque observation et la moyenne de la classe associée [Babu and Murty, 1993]. D'autres articles développent des procédés afin de déterminer des valeurs initiales menant à une meilleure classification, cependant nous n'aborderons pas ici ces travaux.
Cet algorithme permet de trouver rapidement une discrétisation pour un nombre de classe fixé, il représente un cas particulier des modèles de mélange [Celeux and Govaert, 1992] qui peut également être utilisé pour la discrétisation.

Modèles de mélange Les modèles de mélanges sont des modèles statistiques où les observations sont supposées avoir été générées par une somme pondérée (loi de mélange) de lois de densité probabilistes. La loi normale est souvent utilisée dans ce cas, on parle alors de modèles de mélange gaussien. La recherche d'un modèle de mélange gaussien consiste à estimer pour chaque loi normale de cet ensemble 2 paramètres dans le cas univariable, la moyenne et la variance. Le déroulement de l'algorithme utilisé est similaire à celui des K-means à la différence que l'appartenance d'une observation à chaque classe est définie par une probabilité, l'assignation à une classe unique ne s'effectue qu'à la fin de l'algorithme. Un nombre prédéfini de K lois gaussiennes dont les paramètres sont initialisés aléatoirement est utilisé comme point de départ. A chaque itération, on définit dans un premier temps pour chaque observation la probabilité d'appartenance à chacune des K classes puis on estime à nouveau par maximum de vraisemblance les paramètres de chaque loi normale sachant ces probabilités. L'algorithme s'arrête lorsque la variation des paramètres estimés d'une itération à l'autre est inférieur à un seuil fixé. Tout comme l'algorithme des K-moyennes, la recherche de modèle de mélange est dépendant de son initialisation.
A la différence de l'algorithme des K-moyennes où seul doit être recalculé la moyenne de chaque classe, la recherche de modèle de mélange nécessite en plus l'estimation des variances. Cette estimation est plus complexe mais permet en contre partie la modélisation de lois possédant des variances différentes et ainsi une meilleure caractérisation du modèle à estimer.

2.2.1.2 Méthodes multivariées

Minimisation de l'information mutuelle Afin de conserver au maximum l'information mutuelle entre chaque paire de variable Hartemink [2001] propose une méthode de discrétisation par regroupements successifs des classes. L'algorithme nécessite une discrétisation initiale des données en K_{init} classes avec $K_{init} >> K$ et itère un processus de fusion des classes jusqu'à obtenir la discrétisations en K classes souhaitée. La première étape nécessite le calcul pour chaque variable de l'information mutuelle par paire (IM_p) avec chacune des autres variables en considérant la discrétisation courante en K_a classes. Puis pour chaque variable l'algorithme calcule pour les $K_a - 1$ regroupements possibles de 2 classes successives, les nouvelles IM_p avec chacune des autres variables en utilisant leur

discrétisation courante en K_a classes. Par la suite le regroupement qui minimise la perte d'IM_p est appliqué permettant ainsi de passer d'une discrétisation de K_a à $K_a - 1$ classes.
L'intérêt de cette méthode réside dans son objectif de conserver le maximum d'IM entre chaque paire de variables. Il est également possible de mesurer au fil des regroupements la perte d'information subie et donc de choisir une politique de discrétisation offrant un compromis entre un nombre de classes faible et une dégradation de l'IM limitée. Cependant le fait de considérer toutes les IM_p afin de choisir quel regroupement effectuer n'apparaît pas nécessaire dans le cas de variables corrélées à un faible nombre d'autres variables. En effet vouloir minimiser la perte d'IM_p avec des variables peu corrélées revient à vouloir conserver du bruit. Pour pallier cet inconvénient nous proposons en section 4.1.2 une idée d'algorithme permettant de fournir un ensemble de discrétisations minimisant l'information mutuelle sur un sous ensemble de variables.

D'autres approches développées dans [Friedman, 1996, Monti and Cooper, 1998] consistent à coupler la discrétisation des données et l'apprentissage de la structure. L'apprentissage de la structure s'effectue de manière itérative en alternant une phase de recherche locale classique suivit d'une phase de discrétisation des données afin de maximiser l'IM entre chaque variable et ses parents dans le graphe courant.

Par ailleurs l'information mutuelle ne constitue pas le seul critère possible afin de guider les discrétisations supervisées ainsi Lustgarten et al. [2011] proposent une méthode de discrétisation, en amont de l'apprentissage de la structure, maximisant un score bayésien similaire au score *BD* que nous avons présenté auparavant.

2.2.2 Choix du nombre de classes

Pour les différentes méthodes de discrétisation décrites précédemment il est nécessaire de définir le nombre K de classes attendues. Ce choix n'est pas trivial, d'autant plus que celui-ci peut varier selon la nature des données, ainsi lors de la discrétisation de données biologiques telles que l'expression d'un gène, le nombre de classe est fréquemment fixé à 3. Différentes méthodes permettent de déterminer une bonne valeur de K, généralement ces méthodes effectuent différentes discrétisations en faisant varier le nombre de classes avant de sélectionner parmi elles la meilleure politique suivant un critère de type *BIC* ou *AIC* [Jain, 2010].

Dans le contexte de l'apprentissage d'un réseau bayésien discret, le nombre de classes revêt une importance supplémentaire car celui-ci définit la taille du domaine des variables et conditionne donc le nombre de paramètres à apprendre pour définir les probabilités conditionnelles (voir Chapitre 1). L'augmentation du nombre de classes K entraîne deux conséquences majeures. La première apparaît lorsque l'algorithme d'apprentissage de la structure du RB pénalise la dimension de celui-ci. La dimension du RB étant directement liée au domaine des variables (voir section 1.3), celle-ci augmente donc avec K, par conséquent lorsque K augmente la structure apprise devient de moins en moins dense. La deuxième conséquence provient de l'estimation même de ces paramètres lorsque le nombre d'observations N est faible. Chacun des paramètres est estimé par comptage du nombre

d'observations correspondant à une configuration donnée pour une variable cible et ses parents. Lorsque K augmente, certaines configurations peuvent n'être observées que rarement voir jamais, dégradant ainsi la robustesse des paramètres estimées. Pour ces raisons il est préférable de limiter le nombre maximum de classes, $K \approx 5$.

Par ailleurs au sein d'un même réseau étudié, le nombre de classes peut varier d'une variable à l'autre au vu de la répartition de leurs observations, ainsi chaque variable est discrétisée de manière adaptative. Nous présenterons en section 4.1.1 une méthode adaptative utilisant ce principe.

2.3 Conclusion

Nous avons parcouru dans ce chapitre les différentes étapes nécessaires à l'apprentissage de la structure d'un réseau bayésien discret. La première étape consiste, dans le cas de données continues, à discrétiser les données de manière "intelligente" afin d'éviter une perte irréversible d'information. Dans un second temps, intervient l'apprentissage de la structure à proprement parlé en utilisant l'une des deux approches présentées précédemment, les méthodes basées sur la détection des indépendances ou celles qui utilisent une fonction de score. Ces deux approches ont souvent été opposées et comparées, l'avantage revenant souvent aux approches à base de score. Cependant la nécessité de traiter des réseaux de grande taille (plusieurs milliers de variables) laisse la place aux méthodes hybrides qui effectuent d'abord une recherche des indépendances afin de réduire l'espace de recherche et qui emploient par la suite un algorithme à base de score. Plutôt que d'opposer les deux approches il convient donc de les combiner.

Chapitre 3

Nouvelle recherche stochastique gloutonne

Sommaire

3.1 Stochastic Greedy Search 52
3.2 Opérateur SWAP 55
3.3 Opérateurs itératifs 58
3.4 Implémentation sous COMET 62
3.4.1 Utilisation de COMET pour l'apprentissage des réseaux bayésiens 63
3.4.2 Sources d'optimisation 65
3.4.3 Inconvénients du langage COMET 65
3.4.4 Analyse critique de notre implémentation 66
3.5 Évaluation expérimentale des opérateurs 68
3.5.1 Méthodes comparées 68
3.5.2 Critères d'évaluation 69
3.5.3 Réseaux et données utilisés 70
3.5.4 Étude expérimentale 71
3.5.4.1 Opérateurs SWAP et itératifs 71
3.5.4.2 Comparaison avec d'autres approches 75
3.6 Conclusion 83

Nous effectuons dans ce chapitre plusieurs propositions afin d'améliorer l'apprentissage de la structure d'un réseau bayésien dans le cadre des approches à base de score parcourant l'espace des DAG. Nous présentons dans un premier temps une extension de l'algorithme *Greedy Search* permettant de choisir de manière stochastique l'orientation des arcs menant à des structures équivalentes. Dans un second temps nous proposons deux nouveaux opérateurs permettant d'échapper à des situations de minimum local lors de la recherche. Le premier d'entre eux est le SWAP, qui permet de changer un des parents d'une variable en une seule opération. Le second représente une extension des opérateurs classiques afin de considérer des opérations créant des circuits sous condition de pouvoir restaurer l'acyclicité de la structure grâce à un ensemble d'opérations choisies. Puis nous comparons ces différentes propositions afin d'en valider la pertinence face à deux approches de l'état de l'art.

3.1 Stochastic Greedy Search

Présentation de l'algorithme Nous avons déjà abordé précédemment la nécessité d'effectuer un choix lors de l'ajout d'un arc dont l'orientation mène à deux structures équivalentes dans l'espace des DAG. Généralement ce choix est fixé dans l'implémentation même des algorithmes les rendant de fait déterministes (GS, K2, LAGD...). Pourtant l'orientation d'un arc peut modifier la suite de la recherche menant ainsi à différents optima. L'espace des cpDAG offre une solution à ce problème mais complexifie la définition du voisinage. Une alternative consiste à effectuer un choix aléatoire dans l'orientation de ces arcs en maintenant la recherche dans l'espace des DAG. Cette source d'aléa peut être utilisée par toute méthode qui parcourt l'espace des DAG. Afin de pouvoir analyser facilement l'impact des évolutions proposées, nous avons choisi d'utiliser l'algorithme *greedy search*. Par ailleurs l'algorithme GS a souvent montré des résultats honorables dans les comparaisons en dépit de sa simplicité et son principe est souvent repris comme base pour des approches plus complexes.

Nous présentons l'algorithme SGS [1], pour *Stochastic Greedy Search* (Algorithme1) qui utilise cette idée. SGS est basé sur *r* répétitions de l'algorithme GS (ligne 1), la fonction `InitialiseStructure` permet de fournir un graphe initial à chaque exécution de GS. Cette initialisation représente une première source d'aléa, cependant nous verrons qu'il est préférable d'utiliser le graphe vide comme point de départ pour chacune des exécutions de GS. L'algorithme GS est lui même composé d'une répétition de deux phases : exploration du voisinage (ligne 3) puis application de la meilleure opération (ou l'une des meilleures lorsqu'il y en existe plusieurs) (lignes 4, 5) tant que celle-ci améliore le score courant (ligne 2). C'est lors de cette deuxième phase qu'intervient la seconde source d'aléa. La fonction `SelectionAleatoire` retourne une opération choisie aléatoirement parmi l'ensemble des opérations optimales en terme de score. On peut noter que cet ensemble est généralement composé au plus de deux opérations correspondant aux deux orientations pour l'ajout d'un arc, lorsque celle-ci n'intervient pas dans un *v-structure* ni dans la création d'un circuit. Au final la meilleure structure atteinte au bout des *r* exécutions de GS est retournée par l'algorithme (ligne 6).

Cet algorithme se rapproche d'une recherche dans l'espace des cpDAG, à chaque itération la recherche se déplace d'une instanciation aléatoire du cpDAG courant vers une autre instanciation aléatoire d'un cpDAG voisin. Cependant l'algorithme ne considère à chaque itération qu'une seule instanciation possible, ainsi afin de couvrir les différentes configurations d'orientations possibles il est nécessaire d'effectuer un nombre de répétition *r* suffisant.

L'algorithme SGS s'inscrit ainsi parmi les approches stochastiques, nous noterons cependant que cette idée de recherche gloutonne stochastique n'est pas nouvelle et que certaines implémentations de GS effectuent déjà un choix aléatoire parmi l'ensemble des opérations optimales [Murphy, 2001]. Malgré tout, la prise en compte de cet aléa est bien souvent négligée dans la littérature, au point de disparaitre de certaines implémentations disponibles, rendant alors l'algorithme déterministe. Dans ce cas, seule l'initialisation de la recherche permet d'obtenir une variabilité des réseaux appris. De

1. Spirtes and Glymour [1991] ont déjà proposé un algorithme d'apprentissage de RB appelé SGS, cependant cette méthode utilise des tests d'indépendance plutôt qu'une fonction de score.

Algorithm 1: Algorithme *Stochastic Greedy Search*.

Input : ensemble d'observations D, fonction de score f, nombre d'itération r
Output : DAG
$G^* \leftarrow \emptyset$ */* Initialisation du meilleur DAG par le graphe vide*/* ;
$s^* \leftarrow f(G^*, D)$ */* Score du meilleur DAG */* ;
/ Répétition de r recherches gloutonnes aléatoires*/* ;
1 **for** $i \leftarrow 1$ **to** r **do**
  $G \leftarrow$`InitialiseStructure()` */* Choix d'un graphe initial */* ;
  $s \leftarrow f(G, D)$;
2 **repeat**
    améliore $\leftarrow$ **faux** ;
3 $s^{max} \leftarrow \max_{G' \in \texttt{Voisinage}(G)} f(G', D)$;
    if $s^{max} > s$ **then**
      / Sélection aléatoire parmi les meilleurs voisins */* ;
4 $\mathbf{G}^{max} \leftarrow \{G' \in \texttt{Voisinage}(G) | f(G', D) = s^{max}\}$;
5 $G \leftarrow$`SelectionAleatoire(`$\boldsymbol{G}^{max}$`)` ;
      $s \leftarrow s^{max}$;
      améliore $\leftarrow$ **vrai** ;
  until *améliore* = **faux** ;
  / Conserve le meilleur DAG rencontré durant les r recherches */* ;
6 **if** $s > s^*$ **then**
    $G^* \leftarrow G$;
    $s^* \leftarrow s$;
return G^* ;

plus, à notre connaissance, aucune étude n'a encore analysé l'impact de cet aléa sur la qualité de l'apprentissage, ni mis celui-ci en perspective avec une recherche se déroulant dans l'espace des cpDAG.

Complexité de l'algorithme Tout comme l'algorithme GS sur lequel notre algorithme se base, la complexité théorique de SGS est exponentielle. Dans le pire des cas le nombre de graphes parcourus est de l'ordre de $O(p^p)$ et pour chacun de ces graphes l'exploration du voisinage est en $O(p^2)$ avec l'utilisation des opérateurs classiques. De même la valeur de r est en espérance égale au nombre de configurations possibles sur l'orientation des p^2 arcs du graphe soit en $O(2^{p^2})$. Cependant deux facteurs modifient fortement cette valeur de r, le premier étant que toutes les orientations possibles ne sont pas indépendantes, la présence de circuits peut ainsi empêcher certaines configurations sur les orientations tout comme la fonction de score utilisée, ces situations réduisent ainsi le nombre de répétitions nécessaire mais restent difficilement quantifiables. Le deuxième facteur provient du fait que plusieurs répétitions peuvent effectuer les mêmes choix dans l'orientation des arcs ce qui cette fois tend à augmenter le nombre de répétitions nécessaire. Ainsi afin de s'assurer d'avoir parcouru les 2^p configurations d'orientations possibles pour p arcs (ce qui représente un graphe de densité 1, proche de ce que l'on observe en pratique) avec une probabilité β, le nombre de répétitions nécessaire est $r = \frac{\log(1-\beta)}{\log(1-\frac{1}{2^p})}$. Théoriquement donc si $p = 50$ et $\beta = 0.95$, on obtient $r = 3.10^{15}$. Cependant cette valeur théorique est largement sur-estimée du fait qu'une majorité des configurations d'orientations des p arcs ne peut être atteinte lors d'une recherche gloutonne.

Nos diverses expérimentations ont permis d'observer à la fois que le nombre de graphes parcourus est de l'ordre de p et que le plus souvent, seul un faible nombre de répétitions r est nécessaire. De plus lorsque le score utilisé est décomposable l'exploration du voisinage n'est en $O(p^2)$ que pour la première exploration, par la suite seul un nombre limité de scores locaux ont besoin d'être recalculés.

Comparaison avec GES Les comportements de SGS et de GES sont comparés sur la Figure **3.1**. Nous pouvons voir qu'il existe toujours une combinaison de choix sur l'orientation des arcs permettant à SGS de retourner la même structure que GES.

D'un point de vue de l'optimalité des méthodes, GES retourne le vrai graphe lorsque les observations sont représentables par un DAG et quand le nombre de ces observations N est suffisamment grand [Chickering and Maxwell, 2002]. Sous ces mêmes conditions SGS nécessite en plus un nombre r suffisant afin de trouver cette même vraie structure, ainsi théoriquement, SGS est optimal lorsque $r = \infty$. Cependant dans le cas général, les observations ne sont ni en adéquation parfaite avec le vrai RB, ni assez nombreuses pour garantir un quelconque caractère d'optimalité.

L'avantage de SGS sur GES réside dans le fait qu'il ne parcourt que l'espace des DAG et que son voisinage est par conséquent plus simple à définir que celui de GES. En effet la taille du voisinage de GES est exponentielle suivant le nombre de variables lorsque le cpDAG associé à la vraie structure contient un grand nombre d'arcs non-orientés comme le remarque Alonso-Barba et al. [2011].

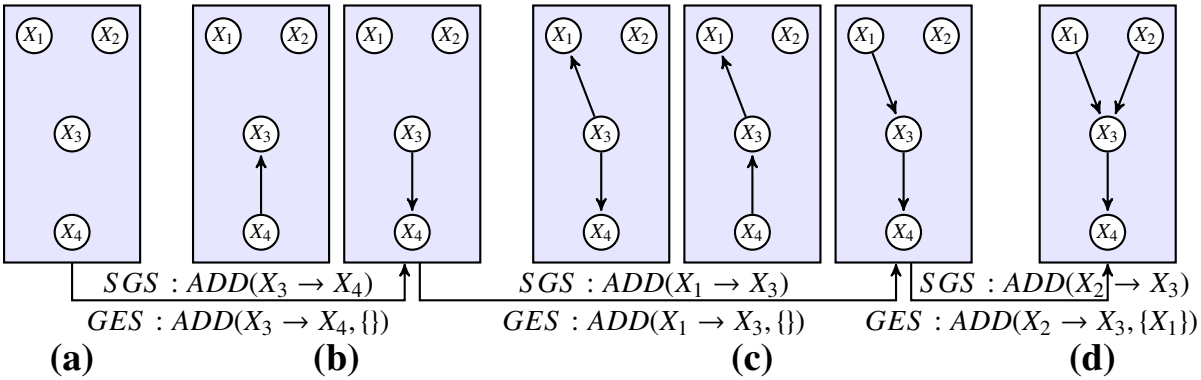


Figure 3.1 – Quatre classes d'équivalence voisines parcourues par GES durant ses 3 premières itérations. (a) GES et SGS débutent du graphe vide (d) Le vrai graphe est trouvé au bout de 3 itérations. L'orientation des arcs $X3 \rightarrow X4$ et $X1 \rightarrow X3$ est choisie aléatoirement par SGS, tandis que GES attend la 3ème itération pour fixer leur orientation dû à la *v-structure* $X1 \rightarrow X3 \leftarrow X2$.

3.2 Opérateur SWAP

Dans des situations d'analyse de données réelles les conditions d'optimalités des algorithmes ne sont que trop rarement réunies notamment celles assurant que les données soient parfaitement représentatives du vrai RB. Ces perturbations peuvent provenir entre autre du bruit inhérent à toute observation physique ou bien aux phases de pré-traitement de ces données comme la discrétisation. On ne peut donc pas se reposer sur des propriétés théoriques d'optimalité au risque d'atteindre des optima locaux de qualité moyenne. Il est nécessaire d'étudier les situations pour lesquelles les opérateurs classiques de déplacement dans l'espace des DAG échouent et de définir de nouveaux opérateurs permettant de se déplacer plus efficacement dans cet espace.

Description de l'opérateur SWAP Nous avons vu précédemment que 3 opérateurs étaient classiquement utilisés afin de se déplacer dans l'espace des DAG : l'ajout d'un arc $X_i \rightarrow X_j$ ($ADD(X_i \rightarrow X_j)$), la suppression d'un arc $X_i \rightarrow X_j$ ($DEL(X_i \rightarrow X_j)$) et l'inversion d'un arc $X_i \rightarrow X_j$ ($REV(X_i \rightarrow X_j)$).

Nous introduisons ici un nouvel opérateur appelé SWAP qui consiste à supprimer un arc $X_i \rightarrow X_j$ tout en ajoutant un nouvel arc $X_k \rightarrow X_j$ ($SWAP(X_i|X_k \rightarrow X_j)$), on modifie ainsi l'ensemble des parents d'une variable cible X_j en remplaçant un des ses parents X_i par une autre variable X_k. Cet opérateur est similaire à l'opération d'inversion correspondant à la suppression d'un arc et à l'ajout de l'arc inverse simultanément.

Afin d'illustrer l'intérêt de l'opérateur SWAP, considérons le graphe courant G_0 décrit par la Figure **3.2**. Supposons que la recherche soit guidée par une fonction de score f, que nous souhaitons maximiser. Ainsi $f(\mathcal{G}, D)$ retourne le score global du graphe $\mathcal{G}$, décrit par la liste des arcs présents, sachant l'ensemble des observations D.

Posons :

$f(\{X_1 \rightarrow X_3\}, D) >$
$f(\{X_2 \rightarrow X_3\}, D) >$
$f(\{X_1 \rightarrow X_3, X_2 \rightarrow X_3\}, D) >$
$f(\{X_3 \rightarrow X_1, X_2 \rightarrow X_3\}, D) >$
$f(\{X_2 \rightarrow X_1, X_2 \rightarrow X_3\}, D) >$
$f(\emptyset, D)$

D'après la fonction de score f la structure optimale correspond au graphe G_1, dans lequel X_1 est parent unique de X_3. Le problème rencontré pour passer de G_0 vers G_1 à l'aide des opérateurs classiques est alors similaire à celui pour lequel l'opération d'inversion est utilisée. Appliquer dans un premier temps $DEL(X_2 \rightarrow X_3)$ ferait décroître le score de même qu'appliquer $ADD(X_1 \rightarrow X_3)$. G_0 représente donc un minimum local si l'on considère uniquement les opérateurs classiques tandis que l'application de l'opération $SWAP(X2|X1 \rightarrow X3)$ permet d'atteindre la structure optimale G_1 sans se soucier du score intermédiaire.

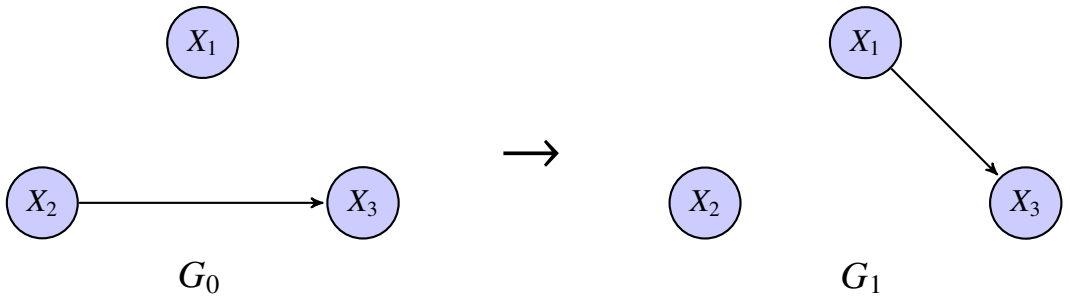


Figure 3.2 – Application de l'opérateur SWAP($X2|X1 \rightarrow X3$).

Cet exemple d'illustration n'est cependant pas valable si la recherche débute du graphe vide puisque G_1 est alors atteignable en appliquant directement $ADD(X_1 \rightarrow X_3)$. On perçoit néanmoins au travers de cet exemple qu'une mauvaise initialisation augmente le risque de tomber dans une situation d'optimum local de faible qualité. Cette observation permet d'expliquer la dégradation des résultats lorsque la recherche est initialisée par un graphe aléatoire. En effet, la probabilité qu'un vrai parent soit présent dans la structure aléatoire initiale pour une variable donnée est faible. Tandis que la probabilité qu'un faux parent soit présent et que celui-ci permettent d'augmenter le score est généralement plus élevée. Ceci vient du fait que l'ensemble des ancêtres et des descendants de la variable considérée dans le vrai graphe, peuvent jouer ce rôle de faux bons parents, si ceux-ci sont suffisamment proches en terme de longueur des chemins et que les probabilités conditionnelles s'y prêtent. Ces situations ne sont donc pas rares et leur nombre augmente avec la connectivité du graphe.

Un second exemple qui peut cette fois-ci se produire lors d'une initialisation par le graphe vide est présenté sur la Figure **3.3**. Dans ce cas le vrai graphe est le DAG G_6 où les deux variables X_3 et

X_4 partagent les mêmes parents X_1 et X_2. Dans ce cas si les distributions de probabilité de X_3 et X_4 sachant X_1 et X_2 sont similaires, alors la corrélation entre X_3 et X_4 est généralement supérieure aux relations de type $parents \rightarrow enfants$ car chacune des variables X_3 et X_4 synthétise l'information des deux parents. Le premier arc ajouté est donc soit $X_3 \rightarrow X_4$ ou $X_4 \rightarrow X_3$ (G_1). Puis les opérations classiques $ADD(X_1 \rightarrow X_3)$ (G_2) puis $ADD(X_2 \rightarrow X_4)$ (G_3) et enfin $ADD(X_2 \rightarrow X_3)$ sont appliquées et mènent à la structure G_4. Afin d'atteindre G_6 à partir de G_4 il est nécessaire d'ajouter $X_1 \rightarrow X_4$ et de supprimer $X_3 \rightarrow X_4$. Ajouter $X_1 \rightarrow X_4$ n'est cependant pas possible dans le cadre d'un score pénalisé car l'information de X_1 est déjà apportée en partie par la variable X_3, le gain en vraisemblance est minime comparé à la pénalité associée à une configuration à 3 parents pour X_4 (graphe G_5). De même commencer par supprimer $X_3 \rightarrow X_4$ dégraderait le score du graphe car X_3 est l'unique parent de X_4 porteur de l'information de X_1. Les opérateurs classiques ne permettent donc pas d'atteindre G_6 à la différence de l'opération $SWAP(X_3|X_1 \rightarrow X_4)$. On notera que $X_1 \rightarrow X_4$ est dorénavant préférée à $X_3 \rightarrow X_4$ du fait de la présence de $X_2 \rightarrow X_4$.

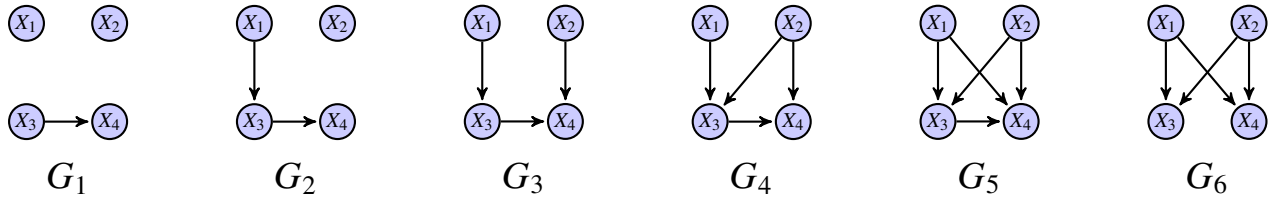


Figure 3.3 – Illustration d'une situation d'optimum local lorsque 2 variables (ici X_3 et X_4) possèdent les même parents.

L'opérateur $SWAP(X_i|X_k \rightarrow X_j)$ permet donc de dépasser certains minima locaux tout en restant local à la seule variable cible X_j, ce qui n'engendre pas de surcoût calculatoire par rapport aux opérateurs classiques. Un tableau récapitulatif des opérateurs que nous utiliserons par la suite est présenté sur le Tableau **3.1** accompagnés de la variation du score que ces opérateurs impliquent.

Opérateurs	$ADD(X_i \rightarrow X_j)$	$DEL(X_i \rightarrow X_j)$	$REV(X_i \rightarrow X_j)$	$SWAP(X_i|X_k \rightarrow X_j)$
Variation du score	$f_j(Pa(X_j) \cup X_i)$ $-f_j(Pa(X_j))$	$f_j(Pa(X_j) \setminus X_i)$ $-f_j(Pa(X_j))$	$f_j(Pa(X_j) \setminus X_i)$ $+f_i(Pa(X_i) \cup X_j)$ $-f_j(Pa(X_j))$ $-f_i(Pa(X_i))$	$f_j(Pa(X_j) \cup X_k \setminus X_i)$ $-f_j(Pa(X_j))$
Effectifs pour p variables et k parents maxi	$p\,(p-k)$	$p\,k$	$p\,k$	$p\,k\,(p-k)$

Tableau 3.1 – Variation du score global induit par l'application des opérateurs locaux et effectifs de chacun d'eux lors de l'exploration du voisinage d'un graphe à p variables et k parents au maximum. $f_j(Pa(X_j))$ représente le score local à la variable X_j sachant ses parents avant l'opération $Pa(X_j)$ et les observations.

Taille des voisinages L'ajout de ce nouvel opérateur augmente de facto la taille du voisinage lors de la recherche. La taille du voisinage d'un graphe à p variables dépend du nombre maximum k de parents pour une variable dans ce graphe. Si l'on décompose le voisinage en fonction du type d'opérateur, on obtient un voisinage borné par $O(p^2)$ pour les opérations d'ajout, $O(p\ k)$ pour les opérations de suppression de même que pour les inversions et $O(k\ p^2)$ pour l'opérateur SWAP. En effet pour chacune des p variables l'opérateur SWAP ne peut être appliqué que sur l'un des k arcs présents que l'on cherche à remplacer par l'un des $p - k \approx p$ arcs absents. L'opérateur SWAP augmente donc sensiblement la taille du voisinage, cependant dans le cas de graphes peu denses, k reste faible (≤ 5), limitant ainsi cette augmentation.

Nous pouvons comparer la complexité de ce nouvel opérateur à deux approches qui augmentent également la taille du voisinage. La première méthode est l'algorithme *LAGD*, cet algorithme effectue une exploration en testant h opérations successives tout en se limitant aux l meilleures opérations à chaque niveau. La taille du voisinage est alors de $O(l^{h-1}p^2)$ où généralement $l \geq 5$ et $h \geq 2$, on notera que la taille du voisinage ne bénéficie pas de la sparsité du graphe à la différence du SWAP. La deuxième méthode dénommée *optimal reinsertion* recherche pour chaque variable l'ensemble optimal de parents (au maximum k parents) ainsi que les enfants. Dans ce cas la taille du voisinage est en $O(C^k_{p-1})$ [Moore and Wong, 2003]. Ces deux méthodes exploitent donc des voisinages plus étendus que celui défini par l'opérateur SWAP ce qui augmente leur complexité.

3.3 Opérateurs itératifs

Nous venons de voir précédemment un nouvel opérateur qui permet d'échapper à certains minima locaux dans le cas d'heuristiques gloutonnes. Cependant une autre restriction fondamentale dans la définition même des réseaux bayésiens est la contrainte d'acyclicité du graphe. Cette contrainte forte interdit certaines opérations même si celles-ci permettent d'augmenter significativement le score local. Pour cela nous avons développé le principe d'opérateurs itératifs dont le but est d'appliquer les opérateurs créant des circuits sous la condition de trouver un ensemble d'opérations associées, qui restaurent l'acyclicité du graphe sans pour autant en dégrader le score global.

Ce principe peut être appliqué à tout opérateur susceptible de créer un circuit, à savoir les opérateurs ADD, REV et SWAP. Dans la suite de cette section nous l'appliquons à l'opérateur SWAP noté SWAP* sachant que le raisonnement est identique pour les opérateurs ADD et REV.

Description de l'opérateur SWAP* Soit un opérateur $SWAP(X_i|X_j \rightarrow X_k)$ dont l'application sur un graphe sans circuit G_0 améliore le score local à X_k et produit un graphe cyclique G_1. On note C l'ensemble des arcs constituant un circuit dans G_1, ainsi, suite à l'application du SWAP, on a notamment $\{X_j \rightarrow X_k\} \in C$. L'opérateur itératif associé $SWAP^\star(X_i|X_j \rightarrow X_k)$ est défini comme l'application au graphe G_0 de l'opérateur initial $SWAP(X_i|X_j \rightarrow X_k)$ et d'un ensemble $\mathcal{L}$ d'opérateurs de suppression $DEL(X_l \rightarrow X_m)$ et de remplacement $SWAP(X_l|X_n \rightarrow X_m)$, tels que $\{X_l \rightarrow X_m \in C\}$, menant ainsi au graphe G_1. $\mathcal{L}$ vérifie alors que le score global de G_1 est supérieur à celui de G_0 et que G_1 ne contient plus aucun circuit (i.e. $C = \emptyset$).

Prenons l'exemple du graphe G_0 de la Figure **3.4** et supposons que l'opération $SWAP(X_2|X_7 \rightarrow X_3)$ est optimale d'après la fonction de score f. Cependant cette opération est interdite dû au circuit $\{X_7 \rightarrow X_3, X_3 \rightarrow X_4, X_4 \rightarrow X_6, X_6 \rightarrow X_7\}$ qui en résulterait. Dans cette situation le principe d'opérateur itératif, noté SWAP*, peut être utilisé. Dans notre exemple, afin d'appliquer l'opérateur SWAP initial, il est nécessaire de supprimer conjointement l'un des 3 arcs du chemin (on ne considère pas l'arc créé par le SWAP initial). Le choix se porte alors sur l'arc le plus faible, c'est-à-dire celui dont la suppression est la moins coûteuse d'après la fonction de score f, dans notre exemple $X_4 \rightarrow X_6$. Cependant si l'application conjuguée de $SWAP(X_2|X_7 \rightarrow X_3)$ et de $DEL(X_4 \rightarrow X_6)$ dégrade le score global du graphe, il est nécessaire de trouver un nouveau parent à X_6 afin de compenser la dégradation du score liée à la suppression d'un de ses parents. On applique alors une opération de SWAP sur X_6 au lieu d'une suppression seule. On sélectionne ainsi le meilleur SWAP portant sur X_6, à savoir $SWAP(X_4|X_5 \rightarrow X_6)$ dans notre cas. Si l'application conjointe des deux opérations $SWAP(X_2|X_7 \rightarrow X_3)$ et $SWAP(X_4|X_5 \rightarrow X_6)$ améliore globalement le score et ne créée pas de circuit alors celles-ci peuvent être validées et appliquées directement. Dans le cas où le score résultant de ces deux SWAP est toujours inférieur au graphe courant, l'opération SWAP* est abandonnée.

Il est cependant possible que l'acyclicité du graphe ne puisse pas être résolue par la suppression ou le SWAP d'un seul arc. La première raison vient du fait que nous supprimons systématiquement l'arc le plus faible appartenant au circuit considéré ce qui ne fourni aucune garantie dans le cas de circuits multiples, c'est-à-dire lorsque qu'il existe plusieurs chemins qui permettent de relier deux variables appartenant au circuit considéré. La seconde raison provient du fait que l'opération de SWAP portant sur X_6 visant à supprimer le circuit courant, peut créer à son tour un nouveau circuit. Dans ces deux situations on recherche un second arc à supprimer (ou à *swapper*) et ce processus est répété jusqu'à la suppression de tout les circuits et tant que le score global du graphe après l'application des différentes opérations ne soit pas inférieur à celui du graphe courant.

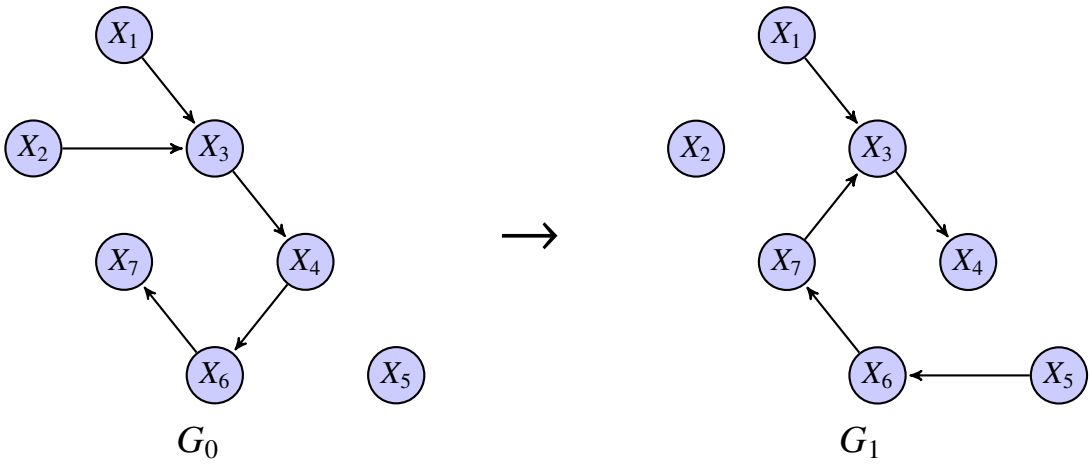


Figure 3.4 – Application de l'opérateur SWAP* afin de casser un circuit en appliquant un second SWAP : $SWAP^\star(X_2|X_7 \rightarrow X_3) = \{SWAP(X_2|X_7 \rightarrow X_3),\ SWAP(X_4|X_5 \rightarrow X_6)\}$.

L'opérateur SWAP* peut donc être vu comme une recherche gloutonne dans l'espace des graphes dirigés cycliques dont le but est d'obtenir, grâce à un ensemble d'opérateurs de suppression et de swap portant sur les arcs les plus faibles des circuits rencontrés, le DAG qui maximise une fonction de score f. L'utilisation d'un SWAP au détriment d'une suppression seule afin de retirer les circuits s'explique par la volonté de ne pas dégrader le score davantage que le gain obtenu grâce à l'opération initiale. Nous pouvons décrire ce processus de façon schématique en considérant que l'opérateur initial cyclique apporte une certaine variation (hypothétique) positive en terme de score. Puis chaque suppression ou SWAP appliqué afin de résoudre les circuits présents réduit cette variation jusqu'à ce que tout les circuits soient résolus ou que la variation devienne négative. La recherche est donc guidée par les valeurs $\Delta_{X_i}(G, OP) = f_{X_i}(G', D) - f_{X_i}(G, D)$ qui représentent la variation du score f localement à la variable X_i lorsque l'opérateur $OP \in \{DEL, SWAP\}$ est appliqué au graphe G, menant ainsi au graphe G'. Ce passage temporaire dans l'espace des graphes cycliques fait perdre tout sens statistique au calcul du score global du graphe courant, les valeurs $\Delta_{X_i}(G, OP)$ représentent alors uniquement des variations hypothétiques du score, celles-ci étant validées dès lors que le graphe final redevient un DAG. Une description de l'opérateur SWAP* est fournie dans l'algorithme 2.

L'amélioration du score produit par l'opérateur SWAP initial est calculée à la ligne 7. Si cette opération est positive, celle-ci est appliquée à une copie G' du graphe courant, de même si aucun circuit n'est créé l'ensemble des opérations à appliquer L ne contient qu'une seule opération ce qui équivaut alors à l'opérateur SWAP classique. Dans le cas contraire il est nécessaire de rechercher une manière de restaurer l'acyclicité du graphe. Tant que le graphe G' est acyclique et que l'application des opérations contenues dans L ne dégrade pas le score global du graphe, on sélectionne l'arc le plus faible en terme de score ($U^* \rightarrow W^*$), qui compose le circuit C détecté grâce à la fonction `ProchainCycle` (ligne 8). Si la suppression de cet arc cumulée avec l'ensemble des opérations présentes dans L ne dégrade pas le score global du graphe (ligne 10), cette nouvelle opération est appliquée à G' et ajoutée à L. Sinon il faut trouver une nouveau parent à la variable W^*, pour cela on sélectionne l'opération de SWAP portant sur W^* qui maximise la fonction de score f (ligne 11). De la même manière que pour la suppression, l'opération SWAP est retenue dans L et appliquée à G' si la variation du score résultante est positive (ligne 12), sinon, l'algorithme s'arrête indiquant qu'il n'existe pas d'ensemble L qui améliore le score global du graphe (ligne 13). Lorsque l'algorithme parvient à résoudre tous les circuits et à améliorer le score du graphe, l'ensemble L est retourné. A noter que Δ contient alors la variation exacte du score global due à l'application des opérations appartenant à L produisant le graphe G'.

L'utilisation de la liste de variables X' à la ligne 9 empêche l'algorithme de restaurer un arc qui a été supprimé lors d'une itération précédente par une opération SWAP (ligne 11), cette condition permet ainsi à l'algorithme de terminer dans tous les cas.

Algorithm 2: Algorithme de l'opérateur $SWAP^{\star}(X|Y \rightarrow Z)$.

Input : opérateur $SWAP(X|Y \rightarrow Z)$, observations D, score f, DAG courant $G(\mathbf{X}, \mathbf{E})$
Output : ensemble d'opérations locales $\mathbf{L}$
$\mathbf{L} \leftarrow \emptyset$ /* *Initialisation à l'ensemble vide des opérations à effectuer* */ ;
$\mathbf{X'} \leftarrow \mathbf{X}$ /* *Parents candidats pour casser le circuit par un SWAP* */ ;
$G' \leftarrow G$ /* *Copie du DAG actuel* */ ;
7 $\Delta = \Delta_Z(G', SWAP(X|Y \rightarrow Z))$ /* *Amélioration due à l'opération initiale* */ ;
if $\Delta > 0$ **then**
  $\mathbf{L} \leftarrow \mathbf{L} \cup \{SWAP(X|Y \rightarrow Z)\}$;
  Applique $SWAP(X|Y \rightarrow Z)$ à G' ;
  /* *Application d'une suppression ou d'un SWAP tant qu'il existe un circuit* */ ;
8 **while** $\Delta > 0 \wedge (\boldsymbol{C} \leftarrow$`ProchainCycle(`$G'$`)`$) \neq \emptyset$ **do**
9 $\mathbf{X'} \leftarrow \mathbf{X'} \setminus nodes(\mathbf{C})$;
  /* *Choisi la meilleur suppression pour casser le circuit* $\boldsymbol{C}$ */ ;
10 $(U^* \rightarrow W^*) \leftarrow argmax_{(U \rightarrow W) \in \mathbf{C} \setminus \{Y \rightarrow Z\}} \Delta_W(G', DEL(U \rightarrow W))$;
  /* *Vérifie si la somme des opérations locales améliore le score* */ ;
  if $\Delta + \Delta_{W^*}(G', DEL(U^* \rightarrow W^*)) > 0$ **then**
    $\mathbf{L} \leftarrow \mathbf{L} \cup \{DEL(U^* \rightarrow W^*)\}$;
    $\Delta \leftarrow \Delta + \Delta_{W^*}(G', DEL(U^* \rightarrow W^*))$;
    Applique $DEL(U^* \rightarrow W^*)$ to G' ;
  else
    /* *Choisi le meilleur SWAP pour augmenter le score* */ ;
11 $(U^*|V^* \rightarrow W^*) \leftarrow argmax_{(U \rightarrow W) \in \mathbf{C}, V \in \mathbf{X'}} \Delta_W(G', SWAP(U|V \rightarrow W))$;
    $\Delta \leftarrow \Delta + \Delta_{W^*}(G', SWAP(U^*|V^* \rightarrow W^*))$;
    if $\Delta > 0$ **then**
      $\mathbf{L} \leftarrow \mathbf{L} \cup \{SWAP(U^*|V^* \rightarrow W^*)\}$;
12 Applique $SWAP(U^*|V^* \rightarrow W^*)$ to G' ;
    else
13 $\mathbf{L} \leftarrow \emptyset$ /* *Annule toutes les opérations* */ ;
return $\mathbf{L}$;

Variantes proposées Différentes variantes de cette approche peuvent être utilisées, notamment pour en réduire la complexité. Il est par exemple possible de fixer un nombre maximum d'opérateurs à appliquer pour casser un circuit bien qu'en pratique la présence de circuits multiples ou la création de nouveaux circuits lorsqu'un SWAP est appliqué (ligne 12 de l'algorithme 2) reste rare du fait de la sparsité des réseaux appris. De même la sélection du premier opérateur de suppression ou de SWAP rencontré qui ne dégrade pas le score global, sans chercher l'opérateur qui le maximise (lignes 10, 11) réduit la complexité du processus.

Plutôt que d'utiliser le score comme seul critère de sélection des arcs à retirer, il est préférable de ne pas supprimer systématiquement l'arc le plus faible mais de favoriser la recherche d'un ensemble minimal d'opérateurs L. Notamment lorsqu'un nombre maximum d'opérateurs pour casser un circuit est fixé. Dans ce cas on recherche en priorité un arc à supprimer qui permet de résoudre à lui seul

l'intégralité des circuits sous condition que celui-ci ne dégrade pas le score global, s'il n'existe pas un tel arc alors le score redevient le critère principal de sélection.

Nous décrivons dans la prochaine section l'implémentation faite des opérateurs itératifs ADD*, REV* et SWAP*. Nous noterons cependant que l'opérateur REV et son extension REV*, deviennent redondants par rapport à l'opérateur ADD*. En effet l'inversion d'un arc $X_i \rightarrow X_j$ équivaut à l'ajout de $X_i \leftarrow X_j$ créant alors un circuit résolu en supprimant $X_i \rightarrow X_j$, ce qui correspond au fonctionnement de l'opérateur ADD*. Autrement dit, pour tout mouvement résultant d'un opérateur REV, il existe un opérateur ADD* représentant ce même mouvement et possédant de fait le même score. Nous conservons cependant ces opérateurs redondants dans l'implémentation décrite par la suite. Du fait du caractère stochastique de SGS tout mouvement résultant d'une inversion pourra donc être réalisé via l'opérateur REV* ou par l'opérateur ADD* correspondant.

3.4 Implémentation sous COMET

L'algorithme SGS, l'opérateur SWAP ainsi que la version itérative des 3 opérateurs ADD, REV et SWAP ont été développés à l'aide du langage de programmation COMET [Michel and Hentenryck, 2002]. Le langage COMET est un langage orienté objet et présente donc de fortes similitudes avec les langages JAVA ou C++, des interfaces sont d'ailleurs disponibles afin de développer des programmes mixtes utilisant des fonctions implémentées en COMET à partir de programmes JAVA ou C++.

Le code développé représente environ 4.000 lignes de code effectives, réparties en 7 classes :

- **AlgoDeRecherche :** implémentation de diverses stratégies d'exploration de l'espace des DAG ;
- **Voisinage :** exploration du voisinage et règles de mise à jour de celui-ci ;
- **Voisin :** données nécessaires afin de définir un voisin d'un graphe ;
- **OrdreTopologique :** ordre topologique des variables du graphe et détection des circuits ;
- **ScoreGénérique :** score global de la structure et mise à jour du score ;
- $BDeu/BIC/fNML$: règle du calcul des différents scores ;

L'organisation de ces classes est présentée sur la Figure **3.5**, ainsi que les attributs et méthodes principales pour chacune d'entre-elles.

Nous présentons dans les sections suivantes les différents points importants liés à notre implémentation sous COMET, en présentant tout d'abord les concepts offerts par ce langage que nous pouvons utiliser pour l'apprentissage d'un réseau bayésien.

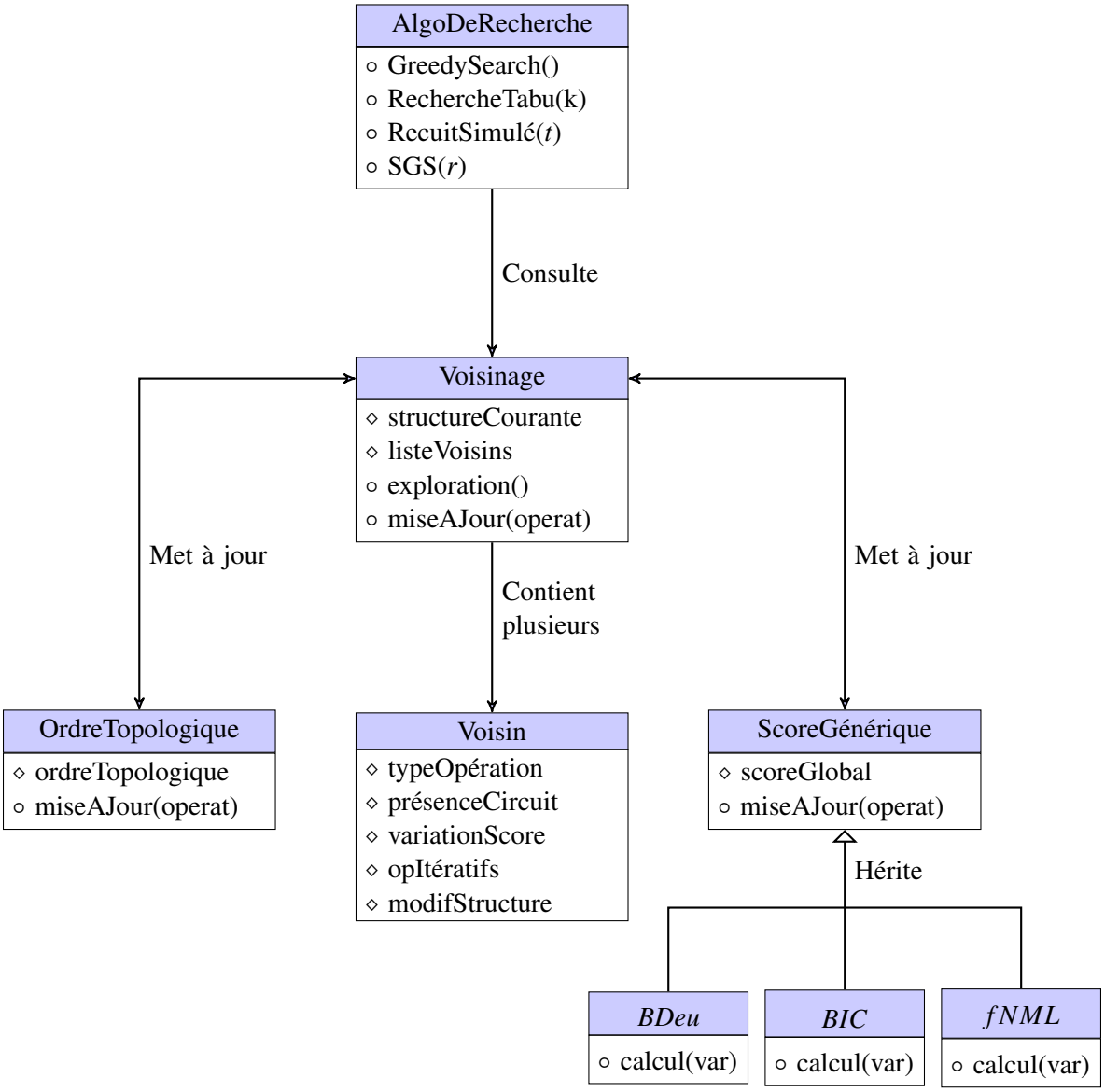


Figure 3.5 – Diagramme de classes présentant les caractéristiques principales (◇ : attributs / ∘ : méthodes) et les interactions entre les différentes classes implémentées en COMET.

3.4.1 Utilisation de COMET pour l'apprentissage des réseaux bayésiens

COMET offre en plus d'un langage classique un ensemble de concepts afin de faciliter le développement d'algorithmes de recherche locale et l'utilisation de la programmation par contraintes. Parmi ces concepts nous pouvons citer les fonctions objectifs, les *invariants* qui dépendent de variables *incrémentales*, les contraintes logiques ou numériques ainsi que la notion de *voisinage*. Certains de ces concepts peuvent être utilisés afin de développer des algorithmes de recherche local pour l'apprentissage de la structure d'un RB. Nous avons notamment utilisé dans notre implémentation

les concepts d'*invariants*, de variables *incrémentales* et de *closure*.

Invariants et variables incrémentales Un *invariant* est une variable dont l'état dépend d'un ensemble de variables dites *incrémentales*, une fois cette relation de dépendance définie, toute modification d'une des variables *incrémentales* met à jour automatiquement l'état de l'*invariant*. Prenons l'exemple d'un invariant Y de type entier dépendant d'un ensemble de k variables entières X_i avec $i \in [1..k]$ tel que $Y = \sum_{i=1}^{k} X_i$. Ainsi toute modification de la valeur d'une des variables X_i mène automatiquement à la mise à jour de la valeur de Y.

Le score global du graphe peut ainsi être défini comme un *invariant* qui dépend de p variables *incrémentales* représentant chacune le score local associé à une variable du graphe. De cette manière le score global est mis à jour automatiquement à chaque changement local de sa structure. De même le score local à une variable donnée, est également un *invariant* qui dépend de l'ensemble des parents de cette variable.

Dans notre implémentation la structure du graphe est représentée de façon compacte comme étant la liste des ensembles de parents de chaque variable. De manière générale la structure du graphe représente la variable incrémentale centrale de la recherche dont dépend à la fois les *invariants* des scores locaux mais aussi un *invariant* pour la détection de circuits. Chaque modification locale de la structure, c'est-à-dire la modification d'un ensemble de parents pour une variable, engendre ainsi automatiquement le re-calcul du score local à cette variable puis le score global du graphe ainsi que la recherche d'un circuit passant par cette même variable.

Un troisième *invariant* lié à la structure du graphe est la définition du voisinage pour chaque variable afin que toute modification de la structure provoque la mise à jour du voisinage des variables concernées. Ce dernier *invariant* est cependant délicat du fait que deux types de modification peuvent être appliqués au graphe. Le premier correspond à l'application de l'opérateur sélectionné afin de faire progresser la recherche, les voisinages sont donc alors automatiquement mis à jour. Le second est appliqué lors de l'exploration du voisinage où les différentes modifications locales possibles ne sont appliquées que temporairement afin de mesurer la variation du score et tester la présence de circuits à la différence des opérations définitives, ces modifications ne doivent pas entraîner une nouvelle exploration du voisinage. Pour cela 2 versions du graphe sont maintenues durant la recherche, l'une représentant le graphe courant sur lequel est appliqué uniquement les opérations définitives provoquant la mise à jour de l'*invariant* lié au voisinage et la seconde utilisée pour tester les opérations dont dépendent les *invariants* liés au calcul du score et à la détection des circuits.

Closure Une *closure* est un ensemble d'instructions encapsulées avec leur environnement, c'est-à-dire l'état des variables au moment où la *closure* est définie, sans être exécutée immédiatement. L'exécution de ces instructions peut ainsi être différée jusqu'à l'appel de la *closure*.

Nous avons utilisé ce système de *closures* afin de définir le voisinage du graphe. Ce voisinage est décomposé localement, pour chaque variable nous retenons l'ensemble des voisins correspondant à l'application d'une opération qui modifie les parents de cette variable. L'ensemble de ces opérations compose le voisinage local de la variable (les opérations d'inversion sont liées aux variables sur lesquelles pointent les arcs du graphe courant).

Chaque voisin est représenté par une structure de données contenant :

- le type d'opération menant à ce voisin,

- la présence d'un circuit et sa composition le cas échéant,

- la variation du score induite par l'opération,

- les opérations pour casser le circuit dans le cas des opérateurs itératifs,

- les instructions correspondant à l'application de l'opérateur contenues dans une *closure*.

Le dernier champ correspond aux instructions nécessaires afin de modifier la structure du graphe courant menant ainsi au graphe associé. Ces modifications engendrent alors la mise à jour des voisinages grâce aux *invariants*. Ces instructions sont donc définies lors de l'exploration du voisinage mais ne sont appliquées que lorsque l'opérateur associé devient le meilleur mouvement possible parmi le voisinage global, conduisant à l'utilisation d'une *closure*.

3.4.2 Sources d'optimisation

En plus des différents concepts offerts par COMET d'autres sources d'optimisation peuvent être exploitées indépendamment du langage employé, pour la recherche. Nous présentons ici deux techniques mises en œuvre dans notre implémentation portant sur les tâches les plus répétitives lors de l'apprentissage d'un réseau bayésien à savoir la détection des circuits et le calcul du score.

Détection des circuits Afin d'améliorer les performances de l'algorithme nous avons utilisé la méthode permettant de maintenir un ordre topologique des variables proposée par Haeupler et al. [2008] dans le codage de la fonction `ProchainCycle`. Cette méthode parcours récursivement lors de chaque ajout d'un arc $X_i \rightarrow X_j$, les ancêtres de X_i et les descendants de X_j afin de déterminer une frontière séparant ces deux variables ce qui permet de restaurer au besoin l'ordre topologique. Lorsque la frontière ne peut être définie, c'est-à-dire lorsqu'une variable est à la fois un ancêtre de X_i et un descendant de X_j, cela indique la présence d'un circuit. Cette technique permet donc de détecter un des circuits dû à l'opération d'ajout et d'obtenir les variables le composant. De plus la recherche s'effectue suivant une politique d'exploration en *largeur d'abord* ce qui assure que les circuits retournés sont de taille minimale.

Utilisation d'un cache Afin d'économiser des appels à la fonction de score, nous avons implémenté un système de cache permettant de retenir pour chaque variable les scores locaux de toutes les combinaisons de k parents rencontrées (avec $k_{max} = 3$). Cette technique coûteuse en terme d'espace mémoire (C_p^k) permet d'améliorer le temps de calcul, le paramètre k_{max} devant être réglé en fonction des caractéristiques du problème.

3.4.3 Inconvénients du langage COMET

Une des principales raisons motivant l'utilisation du langage COMET fut la découverte d'un nouveau langage proposant des concepts utilisables pour l'apprentissage de RB. La découverte d'un

nouveau langage permet de découvrir par la même occasion les lacunes de celui-ci.

A la différence des langages plus généralistes tel JAVA la communauté de développeurs utilisant COMET est plus restreinte limitant de fait la vitesse de développement de nouvelles fonctionnalités. On peut ainsi regretter l'absence de certaines fonctions mathématiques de base comme la fonction *log* ou la fonction Γ utiles pour la définition des fonctions de scores. Pour palier ces manques nous avons choisi de ré-implémenter la fonction *log* directement en COMET et de calculer en dehors de COMET pour la fonction Γ une table des différentes valeurs pouvant être requises durant la recherche. Nous utilisons le fait que la fonction Γ, nécessaire uniquement pour le score *BDeu*, n'est utilisée que pour des arguments discrets qui dépendent à la fois du nombre d'observations pour une configuration $(X_i, Pa(X_i))$ donnée et du nombre de paramètres définissant $\mathbb{P}(X_i|Pa(X_i))$, ces deux termes étant bornés par la définition du problème. Cependant la taille de cette table augmente rapidement avec la dimension des variables, le nombre maximum de parents, ainsi que le nombre d'observations. Les 2 premiers paramètres peuvent malgré tout être limités par l'utilisation d'une discrétisation à faible nombre de classes et la restriction du nombre maximum de parents durant la recherche tandis que le nombre d'observations reste faible dans le cas des données réelles.

De plus ce langage étant interprété avec un mode permettant une compilation à la volée, il est nécessaire d'installer la plate-forme de développement COMET afin d'exécuter tout programme. La licence du logiciel est cependant gratuite mais doit être renouvelée tout les 6 mois.

L'aspect *boite noire* de certains concepts implémentés pose également des difficultés lors de leur utilisation notamment pour corriger des erreurs rencontrées durant l'exécution ou pour analyser l'espace mémoire utilisé ainsi que les temps d'exécution.

3.4.4 Analyse critique de notre implémentation

Les différentes limites de notre implémentation ne sont toutefois pas dues uniquement au langage COMET, certaines proviennent de choix que nous avons effectués. Nous pouvons noter que l'utilisation faite des *closures* et des variables *incrémentales* dans COMET, peut être remplacée par des appels explicites à des fonctions dans le cadre d'un langage plus classique tel que JAVA.

Définition du voisinage Dans notre implémentation nous retenons en permanence pour chaque variable l'intégralité de son voisinage, c'est-à-dire toutes les modifications possibles de son ensemble des parents courant même si ces opérations sont négatives ou engendrent des circuits. Cette décision, coûteuse en terme d'espace mémoire, offre l'avantage d'avoir à tout moment une vision de l'ensemble des opérateurs applicables ainsi que leur impact sur le score et sur l'acyclicité de la structure. Ces informations permettent ainsi de développer des heuristiques de recherche qui n'appliqueraient pas systématiquement le meilleur opérateur et donnent une bonne visibilité sur l'espace dans lequel se déplace la recherche. En outre afin de définir les opérations itératives pour une variable cible, certaines informations déjà calculées liées à des opérations appartenant aux voisinages locaux d'autres variables, peuvent être réutilisées.

Mise à jour du voisinage Maintenir l'ensemble du voisinage à tout moment implique une mise à jour assez lourde à chaque modification dans le graphe. En effet l'ajout d'un arc pour une variable cible peut modifier le caractère acyclique d'opérations inclues dans d'autres voisinages locaux. Afin de réduire en complexité nous limitons le nombre de voisinages à mettre à jour. Considérons l'exemple présenté sur la Figure **3.6**, on remarque que tout ajout d'arcs d'un descendant de X_5 (X_5 inclus) vers un ancêtre de X_3 (X_3 inclus) est interdit à cause des circuits tandis que les arcs des mêmes sources vers X_2 et X_4 sont permis. Après l'opération $SWAP(X_3|X_4 \rightarrow X_5)$ la situation s'inverse, les arcs d'un descendant de X_5 ne peuvent plus aboutir à un ancêtre de X_4 ce qui implique la mise à jour des voisinages locaux des ancêtres de X_3 et X_4. De plus seules les opérations de type $ADD(Descendant(X_5) \rightarrow Ancêtre(X_3) \cup Ancêtre(X_4))$ sont concernées par cette mise à jour. Dans le cas général le voisinage de certains descendants de X_5 doit également être mis à jour lorsqu'il existe dans la structure courante un arc reliant un ancêtre de X_3 ou X_4 vers ce descendant de X_5. On voit donc qu'il est possible de mettre à jour un nombre restreint de voisinage et de façon ciblée sur certaines opérations limitant ainsi le coût de cette mise à jour. Malgré cela la taille des listes des ancêtres et des descendants augmente avec la densité du graphe, ce qui augmente le temps nécessaire à la mise à jour.

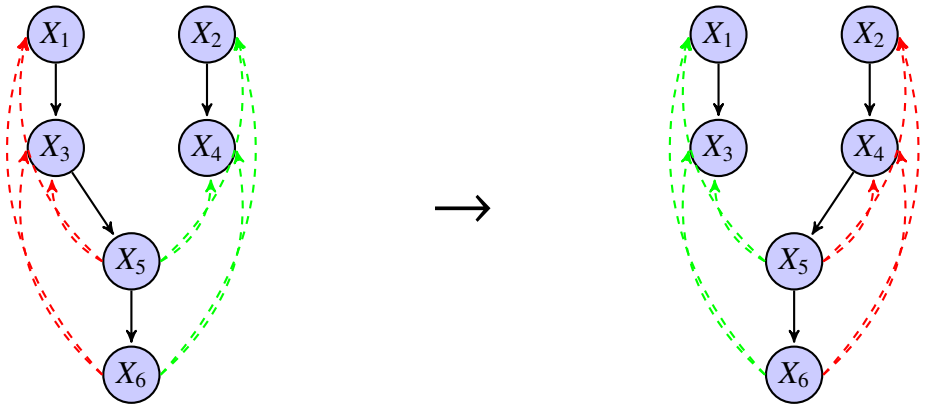


Figure 3.6 – Mise à jour des voisinages locaux après l'application de l'opération $SWAP(X_3|X_4 \rightarrow X_5)$ au graphe initial (à gauche). En traitillé rouge sont représentés quelques arcs potentiels dont l'ajout provoque un circuit, ces opérations appartiennent aux voisinages locaux des variables X_1,X_2,X_3 et X_4. A l'inverse les arcs en traitillé vert indiquent cette fois des ajouts ne créant pas de circuit pour ces mêmes voisinages locaux.

L'utilisation d'opérateurs itératifs ajoute également de nouvelles dépendances entre variables. Reprenons l'exemple de la Figure **3.4** avec l'opération itérative SWAP* définie sur la variable cible X_3 et qui appartient donc au voisinage local de cette variable. Cette opération itérative requiert une seconde opération de SWAP pour la variable X_6 afin de supprimer le circuit créé par le SWAP initial.

Cependant entre l'instant de la recherche où l'opération SWAP* est définie comme étant une opération possible localement à X_3 et celui où l'opération est effectivement appliquée au graphe, plusieurs autres opérations sont généralement appliquées au reste du graphe. Ces opérations peuvent

alors avoir un impact sur l'ensemble des opérations nécessaires à l'opérateur itératif en terme de score ou de consistance, invalidant par conséquent l'opérateur itératif dans son ensemble. Sur notre exemple un second parent peut être ajouté à X_6 ce qui modifierait le score du SWAP portant sur X_6 nécessaire à l'opérateur SWAP*. Il faut donc dans ce cas mettre à jour également le voisinage de X_3. Afin de prendre en compte ce type de mise à jour, chaque voisinage local possède la liste des variables dont le voisinage contient un opérateur itératif dépendant de ce premier voisinage. Ainsi le voisinage local de X_6 contient dans sa liste la variable X_3 afin que le voisinage de cette dernière soit mis à jour si le voisinage de X_6 est modifié.

3.5 Évaluation expérimentale des opérateurs

Afin d'évaluer l'intérêt de l'opérateur SWAP et de la version itérative des opérateurs ADD, REV et SWAP nous effectuons une série de comparaisons sur des benchmarks classiques dans le domaine des réseaux bayésiens. D'autres expérimentations appliquées à des réseaux de gènes seront présentées dans la section 7.2.2.

3.5.1 Méthodes comparées

Dans le but de simplifier la compréhension des résultats nous utilisons une notation indicée de notre algorithme SGS afin d'indiquer les opérateurs utilisés pour définir le voisinage :

- **SGS**^{1} : opérateurs classiques (ADD, DEL, REV) ;
- **SGS**^{2} : opérateurs (ADD, DEL, REV, SWAP) ;
- **SGS**^{3} : opérateurs (ADD*, DEL, REV*, SWAP*) ;

Nous avons comparé les 3 versions de SGS à deux approches à base de score effectuant aussi une recherche gloutonne : l'algorithme GES dont une version est disponible dans le logiciel TETRAD [Scheines et al., 1998] ainsi que l'algorithme LAGD implémenté dans le logiciel WEKA [Hall et al., 2009]. Parmi les critères de choix figure la disponibilité des logiciels gratuitement, le code source modifiable et ayant montré de bonnes performances face aux autres approches de reconstruction de RB.

Le score utilisé par toutes les méthodes est le score *BDeu* avec $\alpha = 1$. Nous n'étudions pas dans ce comparatif l'impact du score choisi ni celui de la valeur de son paramètre α. Nous fixons également un *a priori* uniforme sur l'ensemble des graphes.

3.5.2 Critères d'évaluation

Deux types de critère peuvent être utilisés afin de comparer les performances dans l'apprentissage des RB.

Critères sur la structure La première catégorie mesure uniquement la qualité structurelle du réseau appris (G_{app}) par rapport au vrai graphe (G_{vrai}) en classant chaque arc possible en 4 catégories :

- **Vrai Positif** (VP) : arc présent à la fois dans G_{vrai} et G_{app} ;
- **Faux Positif** (FP) : arc présent dans G_{app} mais absent de G_{vrai} ;
- **Vrai Négatif** (VN) : arc absent à la fois de G_{vrai} et G_{app} ;
- **Faux Négatif** (FN) : arc absent de G_{app} mais présent dans G_{vrai} ;

Ces 4 mesures permettent de définir différents critères parmi lesquels nous pouvons citer :

- **Précision** $= \frac{VP}{VP+FP}$, indique le nombre d'arcs corrects par rapport au nombre d'arcs appris ;
- **Sensibilité** $= \frac{VP}{VP+FN}$, indique le nombre d'arcs corrects par rapport au nombre de vrais arcs à apprendre ;
- **Spécificité** $= \frac{VN}{VN+FP}$, indique le nombre de non-arcs corrects par rapport au nombre de vrais non-arcs à apprendre ;
- **Distance d'édition** $= FP + FN$, indique le nombre de changements d'arc à effectuer pour passer de G_{app} à G_{vrai} ;

Le défaut majeur de ces critères apparaît lorsqu'il s'agit de comparer des méthodes qui fournissent des réseaux avec différents niveaux d'orientation. Dans notre comparatif l'algorithme GES produit un cpDAG tandis que LAGD et SGS fournissent des DAG. Dans ce cas plusieurs solutions sont possibles, comme sélectionner l'instanciation du cpDAG minimisant le critère d'évaluation choisi ou bien effectuer le raisonnement inverse en recherchant le cpDAG correspondant aux DAG appris par les autre méthodes puis en adaptant la définition des 4 catégories d'arc. Nous avons choisi une autre alternative en comparant les versions non-orientées des graphes (leur squelette). Dans ce cas nous comparons des arêtes et non plus des arcs, la matrice d'adjacence des nœuds du graphe peut ainsi être symétrisée et la comparaison s'effectue alors uniquement sur la demi-matrice.

Toutes les mesures de qualités structurelles que nous utiliserons par la suite sont donc basées sur la comparaison du squelette du RB appris avec celui du vrai RB.

Critères sur la structure et les paramètres En plus de la qualité structurelle du graphe appris, un second type de critère d'évaluation tient compte des paramètres du RB c'est-à-dire des probabilités conditionnelles estimées sachant la structure et les observations. Dans ce cas une première possibilité consiste à utiliser le score guidant la recherche comme mesure de qualité, dans un but de partialité il

est alors préférable de restreindre la comparaison aux algorithmes utilisant ce score. L'inconvénient de ce type de critère provient du fait que le score atteint par l'heuristique puisse dépasser le score du vrai graphe. Dans cette situation on dira que le réseau reconstruit *sur-apprend* les données, un phénomène difficile à contrôler *a priori* surtout quand le nombre d'observations est faible. L'une des alternatives repose sur la divergence de *Kullback-Leibler* afin de comparer la différence entre les deux distributions représentées par le vrai RB et le RB appris. Plus cette distance est faible, plus les deux distributions et donc les deux réseaux sont proches.

Dans le cadre de l'apprentissage d'un RB, les critères mesurant la qualité des algorithmes à partir de la structure et des paramètres semblent les plus adaptés. Cependant dans le cas d'une application à un phénomène physique, le vrai réseau n'est pas modélisable exactement par un RB ce qui rend l'utilisation d'une mesure comme la divergence de *Kullback-Leibler* peu judicieuse. Dans les expérimentations présentées par la suite nous comparons les méthodes à la fois sur le score maximal atteint afin de mesurer leur capacité à se déplacer dans l'espace de recherche ainsi que sur les critères liés uniquement à la structure du graphe.

3.5.3 Réseaux et données utilisés

Nous utilisons 4 RB classiquement utilisés dans la littérature dont quelques unes des caractéristiques sont fournies dans le Tableau **3.2**. La description de ces réseaux (structure et paramètres) est disponible à l'adresse suivante http://www.cs.huji.ac.il/site//labs/compbio/Repository/.

Tableau 3.2 – Propriétés des 4 réseaux bayésiens utilisés.

	ALARM	INSURANCE	HAILFINDER	PIGS
Noeuds	37	27	56	441
Arcs	46	52	66	592
Degré entrant max	4	3	4	2
Tailles des domaines	2-4	2-5	2-11	3-3
Chemin le plus long	11	10	14	6

Pour chacun de ces 4 réseaux nous avons fait varier le nombre d'observations entre 50, 500 et 5000. Afin d'obtenir des statistiques significatives pour cette comparaison, 100 jeux de données pour chaque réseau et chaque nombre d'observations ont été générés. Ces données ont été obtenues à l'aide du programme Maltab *Causal Explorer* [Aliferis et al., 2003b]. Nous utiliserons dans la section suivante ces données dans différentes configurations afin d'analyser le comportement de nos opérateurs.

3.5.4 Étude expérimentale

3.5.4.1 Opérateurs SWAP et itératifs

Dans un premier temps nous étudions le comportement des 3 versions de SGS sur le réseau ALARM pour différents jeux de données composés de 500 observations.

Influence de l'initialisation Nous observons ici l'influence de l'initialisation sur l'algorithme SGS, nous nous restreignons dans ce cas à un seul jeu de données composé de 500 observations. Nous effectuons 1000 exécutions des 3 versions de SGS (nombre de répétitions $r = 1000$) à partir d'un graphe vide ainsi qu'à partir de 1000 DAG aléatoires contenant 71 arcs tirés suivant une loi uniforme sur l'ensemble des arcs possibles ne produisant pas de cycles. Les scores atteints par chaque exécution sont représentés sur les 2 histogrammes de la Figure **3.7**.

Dans le cas d'une initialisation par le graphe vide (histogramme du haut), nous observons clairement le caractère stochastique des algorithmes avec une variation du score *BDeu* uniquement due aux choix sur l'orientation des arcs. L'opérateur SWAP n'apporte, sur ce jeu de données, aucune amélioration notable par rapport aux opérateurs classiques (SGS^1 vs SGS^2) tandis que les opérateurs itératifs (SGS^3) améliorent sensiblement les performances en restreignant la variance des scores atteints. Les opérateurs itératifs corrigent ainsi l'effet dû à une mauvaise orientation de certains arcs, leur permettant d'atteindre une solution de bonne qualité indépendamment des orientations choisies.

Ces observations se confirment lorsque la recherche est initialisée à partir d'un graphe aléatoire. Les résultats en moyenne de SGS^1 et SGS^2 se dégradent ce qui montre que les opérateurs classiques ainsi que le SWAP ne permettent pas à l'algorithme de s'écarter suffisamment du graphe initial, qui converge alors vers un minimum local de faible qualité. Les opérateurs itératifs utilisent au contraire ce graphe initial afin d'atteindre des graphes de meilleur qualité que lors d'une initialisation par le graphe vide, même si la variance des scores atteints augmente, la moyenne des scores ainsi que le score maximal est en hausse. SGS^3 tire ainsi pleinement partie des bons arcs fournis lors de l'initialisation tout en corrigeant les mauvais arcs grâce aux nouveaux mouvements permis par les opérateurs itératifs.

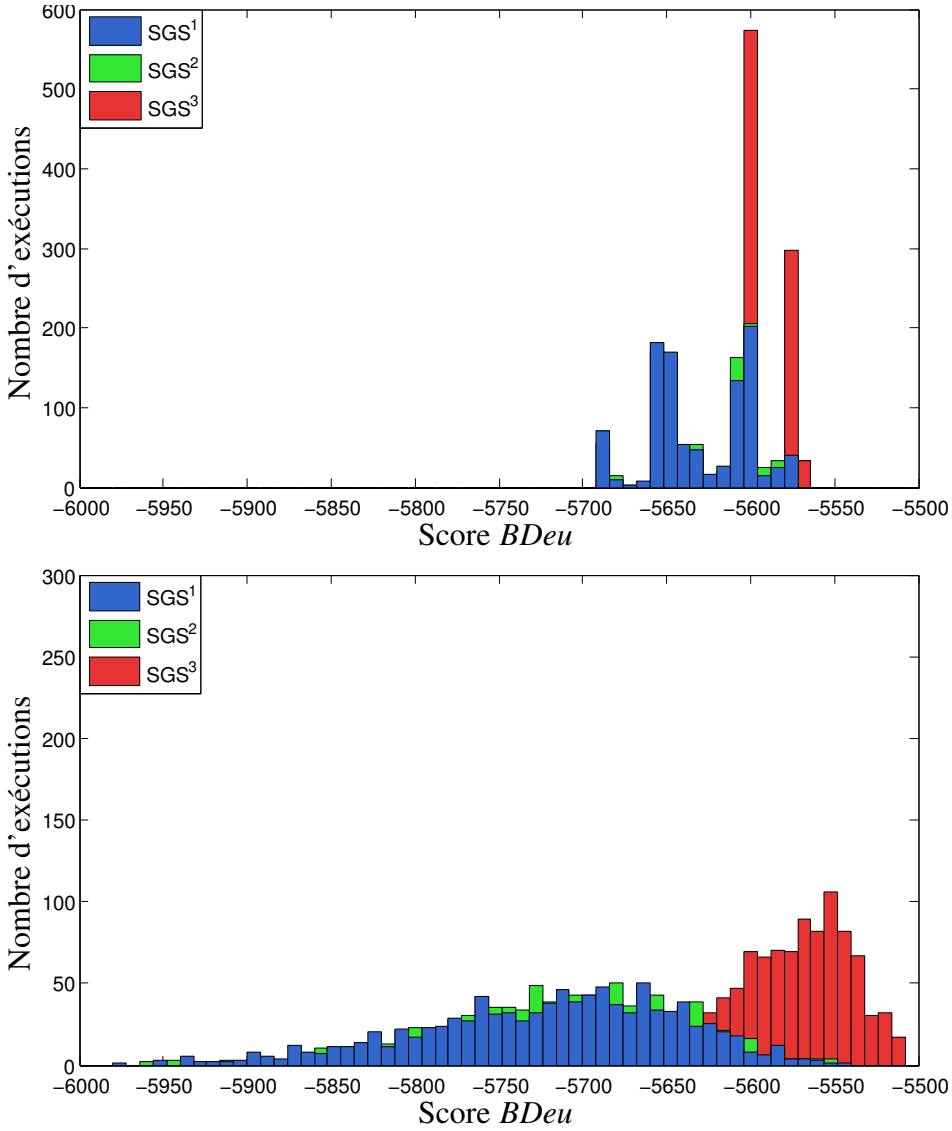


Figure 3.7 – Score *BDeu* atteint par les 1000 exécutions de SGS[1] (en bleu), SGS[2] (en vert) et SGS[3] (en rouge) à partir du graphe vide (histogramme du haut) et à partir de graphes aléatoires (en bas) pour un jeu de 500 observations du réseau ALARM.

Applications des opérateurs Nous nous restreignons cette fois à une seule exécution pour un seul jeu de données de 500 observations afin d'analyser l'évolution du nombre d'opérateurs appliqués durant la recherche en fonction des 2 initialisations possibles (graphe vide ou aléatoire) pour les 3 versions de SGS. Les 6 exécutions sont représentées sur la Figure **3.8**.

Dans le cas d'une exécution à partir du graphe vide, le comportement des 3 algorithmes est similaire avec un nombre d'ajouts prédominant. Pour SGS[1] et SGS[2] le nombre d'arcs remis en cause

est quasi-nul, seules une inversion et une suppression sont effectuées pour SGS^1 et un seul SWAP ainsi qu'une inversion pour SGS^2. Ces deux algorithmes se dirigent rapidement vers une solution sans remettre en cause les premiers arcs ajoutés. Seul SGS^3 applique plus d'une dizaine d'opérateurs itératifs ($ADD^\star$ et $SWAP^\star$) ce qui lui permet d'atteindre la distance d'édition la plus faible des 3 algorithmes malgré l'augmentation de cette distance lors de l'ajout des derniers arcs. On remarque par ailleurs que le premier opérateur itératif est appliqué dès la $30^{\text{ème}}$ itération alors même que la connectivité du graphe est faible (<0.5) ce qui montre que la contrainte d'acyclicité restreint très tôt le déroulement de la recherche.

La situation diffère lorsque l'initialisation se fait à partir d'une structure aléatoire. Bien que les premiers opérateurs appliqués par SGS^1 soient toujours des ajouts d'arcs, le nombre de suppressions augmente rapidement afin d'être finalement l'opérateur le plus utilisé, le nombre d'inversions reste quant à lui très faible. Dans le cas de SGS^2 et SGS^3, l'opérateur SWAP est le plus employé permettant d'effectuer une suppression et un ajout lors d'une même itération ce qui réduit le nombre d'itérations nécessaires et permet de faire diminuer la distance d'édition plus rapidement. La version itérative de l'opérateur SWAP est par ailleurs employée fréquemment par SGS^3 et cela dès les premières itérations, ce qui permet à nouveau d'obtenir la distance d'édition la plus faible des 3 algorithmes. Tout les arcs pour lesquels aucun SWAP n'est possible sont finalement supprimés, ce qui explique l'application de l'opérateur DEL lorsque le nombre de SWAP se stabilise durant les dernières itérations.

Pour ces différentes configurations on peut remarquer que la distance d'édition n'est pas strictement décroissante alors même que le caractère glouton de l'algorithme assure que le score du graphe est strictement croissant en fonction du nombre d'itérations. Nous pointons ici grâce à ces premiers résultats l'un des problèmes récurrents lors de l'apprentissage de la structure à savoir le nombre limité d'observations. Nous noterons cependant que le nombre d'observations utilisées ici ($n = 500$) représente un nombre d'observations élevé dans le cas de données réelles. Nous sommes donc dans une situation où l'optimisation de la fonction de score ne mène pas systématiquement à une amélioration structurelle du graphe.

Impact du nombre de répétitions r Comme indiqué précédemment une borne théorique au nombre d'exécutions de SGS nécessaire afin de parcourir l'ensemble des orientations possibles avec une probabilité fixée est extrêmement élevée, même pour quelques dizaines de variables. Ces orientations ne sont cependant pas toutes indépendantes entre elles ce qui réduit le nombre de ces configurations. Nous observons sur la Figure **3.9** l'évolution du meilleur score atteint, en moyenne sur 30 jeux de données différents composés de 500 observations, par SGS lorsque le nombre d'exécutions augmente ($1 < r < 50$) à partir du graphe vide ou d'un DAG aléatoire obtenus de la même manière que précédemment. Dans un but de comparaison nous traçons également la moyenne des scores atteints sur ces 30 jeux de données par GES ainsi que celle obtenue par le vrai graphe.

Initialisation par le graphe vide *Initialisation par un graphe aléatoire*

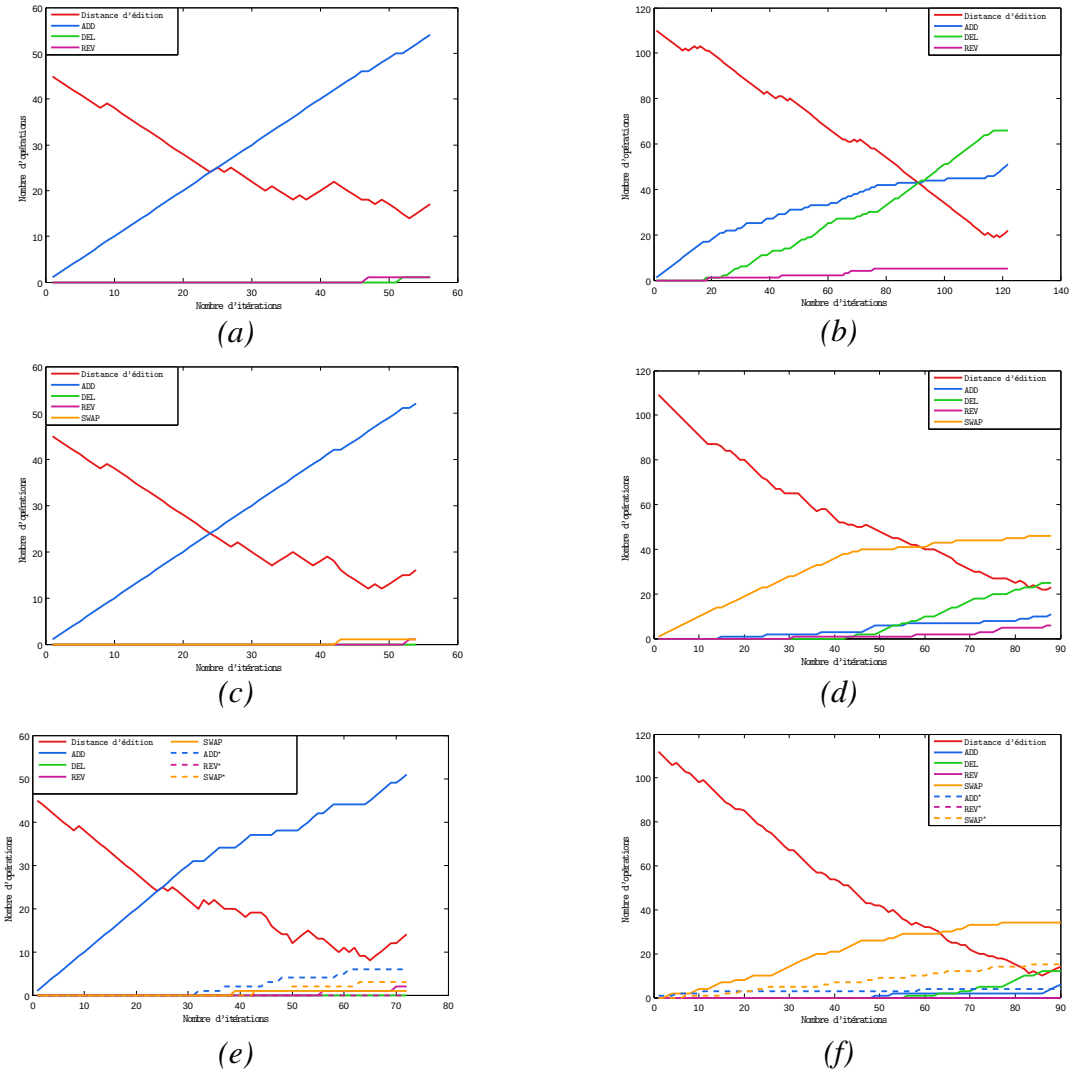


Figure 3.8 – Évolution de la distance d'édition et du nombre d'opérateurs appliqués lors d'une exécution de SGS^1 (*a, b*), SGS^2 (*c, d*) et SGS^3 (*e, f*) à partir du graphe vide (*a, c, e*) ou d'un DAG aléatoire (*b, d, f*). Pour toutes les figures sont indiqués : la distance d'édition (en rouge continu), le nombre d'opérateurs ADD (en bleu continu), DEL (en vert continu) et REV (en rose continu), puis spécifiquement à SGS^2 et SGS^3 le nombre de SWAP (en orange continu) puis les opérateurs itératifs uniquement pour SGS^3 : ADD* (en bleu traitillé), REV* (en rose traitillé) et SWAP* (en orange traitillé).

A partir du graphe vide, les scores moyens atteints par les différentes versions de SGS se stabilisent rapidement dès 10 répétitions. SGS^3 se démarque en dépassant sensiblement SGS^2 et SGS^1 de même que le score de la vraie structure. SGS^2 se montre par ailleurs plus performant en moyenne que SGS^1 sur les 30 jeux de données considérés à la différence des observations précédentes réalisées sur un seul

jeu d'observations.

Comme observé avec les histogrammes de la Figure **3.7**, l'initialisation par le graphe aléatoire permet d'améliorer le score atteint par SGS^3 et dégrade ceux atteints par SGS^1 et SGS^2. On notera que l'amélioration brutale observée pour ces deux versions de SGS à la $27^{ème}$ répétition a pour origine la bonne qualité du graphe initial ce qui oriente favorablement la suite de la recherche. De manière générale toutes les versions de SGS indépendamment de leurs initialisations obtiennent un score plus élevé que GES.

Il est à noter que la moyenne optimale pour ces 30 jeux de données est de -5490.23 avec un nombre de parents maximum fixé à 4, cette valeur a été obtenue via un échange personnel avec James Cussens qui a développé une méthode de recherche exacte inspirée des travaux de Jaakkola et al. [2010]. On remarque ainsi que le score atteint par la meilleure configuration de SGS est proche de cet optimum.

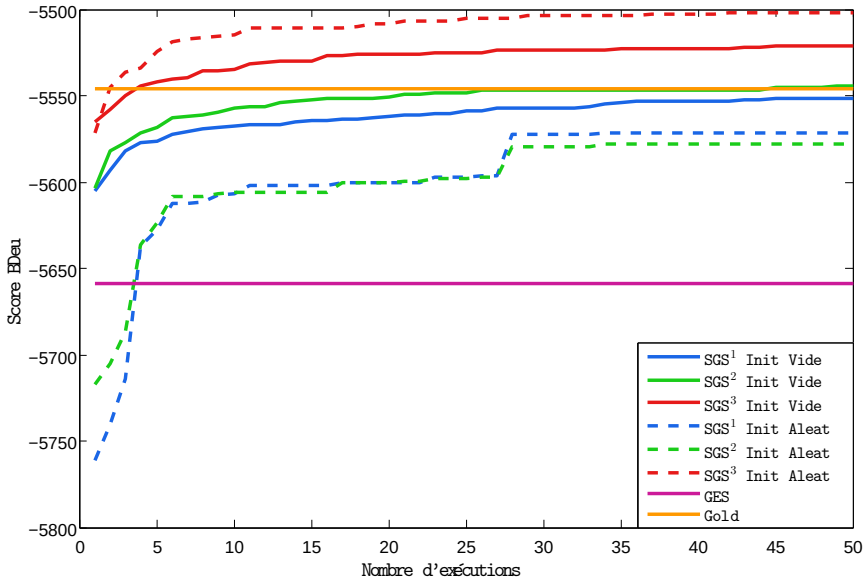


Figure 3.9 – Meilleur score *BDeu* atteint en moyenne sur les 30 jeux de données pour SGS^1(en bleu), SGS^2(en vert), SGS^3 (en rouge) en fonction du nombre de répétitions r. Pour SGS l'initialisation s'effectue soit à partir du graphe vide (en ligne continue) ou à partir de DAG aléatoires (en pointillé). Les scores moyens atteint par GES (en rose) ainsi que celui du vrai graphe (en orange) sont également indiqués.

3.5.4.2 Comparaison avec d'autres approches

Nous comparons dans cette section les 3 versions de SGS aux algorithmes GES et LAGD sur les 4 réseaux benchmark en faisant varier le nombre d'observations entre 50 et 5 000.

Réglage des méthodes Les méthodes utilisées ne nécessitent pas toutes le même nombre de paramètres. L'algorithme GES ne requiert ainsi aucun réglage (hormis le paramètre α pour le score

BDeu), son exécution étant totalement déterministe à la différence des deux autres méthodes. Les algorithmes SGS nécessitent quant à eux de fixer un nombre de répétitions r afin d'explorer les différentes orientations possibles et de conserver le DAG obtenu qui maximise le score, dans notre cas le score *BDeu*. Malgré le fait que l'algorithme LAGD soit déterministe par défaut du fait qu'il choisisse les orientations suivant l'ordre d'apparition des variables dans le fichier des observations, cet ordre peut être modifié afin de lui adjoindre un caractère stochastique qui influera sur l'orientation des arcs. Cette modification est motivée par l'observation que cet ordre impacte fortement les performances de l'algorithme, de la même manière que pour SGS nous avons donc exécuté LAGD r fois en utilisant pour chaque exécution un ordre aléatoire des variables puis avons conservé le meilleur DAG atteint en terme de score. LAGD nécessite deux paramètres supplémentaires à savoir le nombre d'opérateurs consécutifs k appliqués à chaque itération ainsi que le nombre de directions l dans lequel l'algorithme effectue les k opérations.

Dans les expérimentations suivantes nous avons fixé comme valeurs de paramètres : $r = 10$ pour SGS ainsi que pour LAGD sur les 3 réseaux ALARM, INSURANCE et HAILFINDER. Pour ce dernier les deux paramètres spécifiques ont été réglés suivant l'article de Holland et al. [2008] à savoir $k = 5$ et $l = 2$. Dans le cas du réseau PIGS l'algorithme LAGD n'a pas pu fournir de résultats dans un temps raisonnable et nous avons dû restreindre à une seule répétition les algorithmes SGS ($r = 1$) pour cette même raison, liée au nombre important de variables constituant le réseau.

Nous avons également dû limiter le nombre maximum de parents autorisés durant la recherche et avons adapté cette valeur dans certains cas spécifiques. Ainsi le nombre de parents maximum est fixé à 5 pour l'algorithme SGS et LAGD indépendamment du réseau tandis que l'algorithme GES est limité respectivement à 7 et 10 parents pour les réseaux HAILFINDER et PIGS tandis qu'aucune limite n'est imposée pour les 2 autres réseaux. Bien que les limites de GES paraissent plus élevées que pour les deux autres approches, celles-ci sont nécessaires afin de permettre à l'algorithme d'ajouter suffisamment d'arcs durant sa phase *backward* pour représenter au minimum toutes les dépendances décrites par les données.

Le dernier réglage concerne l'initialisation de la recherche. Les résultats précédents ayant montré l'impact positif d'une initialisation par un graphe aléatoire uniquement pour SGS^3 et par soucis d'équité envers tout les algorithmes, le graphe vide servira de point de départ à l'ensemble des méthodes.

Scores atteints Le Tableau **3.3** indique le score *BDeu* moyen atteint sur 100 jeux de données, par les différents algorithmes pour chacun des 4 réseaux benchmark. L'algorithme SGS^3 atteint les meilleurs scores dans 11 cas sur 12, seul l'algorithme GES dans le cas du réseau PIGS avec un grand nombre d'observations se révèle meilleur. Cette situation s'explique en partie du fait que l'algorithme SGS n'est exécuté qu'une seule fois pour le réseau PIGS dégradant ainsi ses performances. De plus la structure particulière de ce réseau favorise les bonnes performances de GES comme nous le montrerons un peu plus tard.

Certains écarts restent toutefois ténus notamment entre SGS^1 et SGS^2. Afin d'analyser plus

précisément ces résultats nous présentons dans le Tableau **3.4** les résultats du test des rangs signés de Wilcoxon [Wilcoxon, 1945] permettant de comparer les méthodes deux à deux sur chacun des 100 jeux de données. Chaque comparaison indique si l'une des deux méthodes est significativement meilleure que l'autre. Dans le cas de comparaisons multiples comme ici, il est nécessaire d'utiliser la correction de Bonferroni afin de maintenir le niveau global de l'ensemble des tests à 5% (i.e. la probabilité de dire à tort qu'une méthode est supérieure à une autre dans le tableau est de 0.05). L'erreur de première espèce de chaque test est ainsi révisée à $\frac{0.05}{4*3*10} = 4.1610^{-4}$ (4 réseaux $\times$ 3 nombres d'observations $\times$ 10 comparaisons deux à deux).

Ces résultats sont résumés dans le Tableau **3.5** où pour chaque méthode est compté le nombre de configurations où celle-ci est significativement meilleure qu'une autre, auquel on retranche le nombre de situations où la méthode est significativement moins bonne, les comparaisons non-significatives étant ignorées.

On observe grâce à ce test statistique que SGS^3 est significativement meilleur dans 9 configurations sur les 12 testées. Les 3 autres configurations correspondent au cas du réseau Pigs évoqué précédemment, ainsi qu'aux résultats peu significatifs sur le réseau Hailfinder avec peu d'observations. SGS^2 arrive en seconde place grâce à de meilleurs résultats sur alarm par rapport à SGS^1. On peut noter la supériorité de SGS^1 sur LAGD probablement due au processus de tirage aléatoire sur l'orientation des arcs équivalents. En effet l'algorithme SGS détermine aléatoirement chaque orientation durant la recherche tandis que LAGD conserve le même ordre pour une exécution complète de l'algorithme, l'ordre des variables n'étant modifié qu'entre chaque répétition. L'algorithme GES obtient les plus mauvais résultats de ce comparatif dû au faible nombre d'observations qui n'assure pas le caractère optimal des 2 phases de l'algorithme.

Tableau 3.3 – Score *BDeu* moyen obtenu par les 3 versions de SGS ainsi que par les algorithmes GES et LAGD sur 100 jeux de données composés de 50, 500 et 5 000 observations pour les 4 réseaux benchmark.

Nombre		Alarm			Insurance	
d'observations	50	500	5 000	50	500	5 000
SGS^1	-654.10	-5 567.78	-51 188.57	-867.89	-7 299.63	-67 061.77
SGS^2	-652.58	-5 557.05	-51 188.66	-867.17	-7 296.85	-67 045.81
SGS^3	**-649.57**	**-5 526.02**	**-51 024.91**	**-865.10**	**-7 279.04**	**-66 962.35**
LAGD	-660.35	-5 563.82	-51 158.67	-871.15	-7 320.04	-67 155.45
GES	-754.37	-5 659.65	-51 260.17	-897.66	-7 513.71	-67 474.78
		Hailfinder			Pigs	
SGS^1	-3 169.29	-27 194.00	-25 1315.0	-21 297.97	-175 939.3	-1 665 655
SGS^2	-3 168.56	-27 193.15	-25 1314.2	-21 220.95	-175 928.2	-1 665 667
SGS^3	**-3 168.25**	**-27 192.50**	**-25 1278.8**	**-21 064.71**	**-175 805.9**	-1 665 142
LAGD	-3 170.04	-27 192.55	-25 1379.5	n/a	n/a	n/a
GES	-3 187.66	-27 923.13	-25 5973.0	-23 368.98	-176 017.8	**-1 664 300**

Tableau 3.4 – Comparaison des méthodes deux à deux en utilisant le test de Wilcoxon (taux erreur = 5%). Pour chaque comparaison *Méthode1 vs Méthode2*, "+" signifie que la *Méthode1* est significativement meilleure que la *Méthode2*, "−" signifie que la *Méthode1* est significativement moins bonne que la *Méthode2* et "∼" signifie qu'aucune des deux méthodes ne se démarque.

	Alarm			Insurance		
Nombre d'observations	50	500	5 000	50	500	5 000
SGS^3 vs SGS^1	+	+	+	+	+	+
SGS^3 vs SGS^2	+	+	+	+	+	+
SGS^2 vs SGS^1	+	+	∼	∼	∼	∼
SGS^3 vs GES	+	+	+	+	+	+
SGS^3 vs LAGD	+	+	+	+	+	+
SGS^2 vs GES	+	+	∼	+	+	+
SGS^2 vs LAGD	+	∼	∼	+	+	+
SGS^1 vs GES	+	+	∼	+	+	+
SGS^1 vs LAGD	+	∼	∼	+	+	+
LAGD vs GES	+	+	∼	+	+	+
	Hailfinder			Pigs		
SGS^3 vs SGS^1	∼	∼	+	+	+	+
SGS^3 vs SGS^2	∼	∼	+	+	+	+
SGS^2 vs SGS^1	∼	∼	∼	+	∼	∼
SGS^3 vs GES	+	+	+	+	+	-
SGS^3 vs LAGD	∼	∼	+	n/a	n/a	n/a
SGS^2 vs GES	+	+	+	+	∼	-
SGS^2 vs LAGD	∼	∼	+	n/a	n/a	n/a
SGS^1 vs GES	+	+	+	+	∼	-
SGS^1 vs LAGD	∼	∼	+	n/a	n/a	n/a
LAGD vs GES	+	+	+	n/a	n/a	n/a

Tableau 3.5 – Résumé des tests de Wilcoxon.

	Alarm	Insurance	Hailfinder	Pigs
SGS^3	12	12	6	7
SGS^2	2	3	3	-2
SGS^1	-2	3	3	-4
LAGD	-3	-6	0	n/a
GES	-9	-12	-12	-1

Distance d'édition L'un des critères utilisés afin de comparer les algorithmes de reconstruction d'un RB uniquement du point de vue de la structure apprise est la distance d'édition entre la structure apprise et le vrai graphe. Comme indiqué précédemment nous comparons uniquement le squelette des RB, cette distance correspond ainsi au nombre d'ajouts des vraies arêtes manquantes et de suppressions des fausses arêtes apprises qui doivent être appliqués au squelette du réseau appris afin d'obtenir celui du vrai graphe. Les distances d'édition moyennes sur les 100 jeux de données sont indiquées dans le Tableau **3.6**.

Une observation générale montre que les distances d'édition ne sont que faiblement corrélées avec le score *BDeu*, ainsi suivant ce critère les meilleurs résultats sont obtenus par GES alors que celui-ci atteint les scores *BDeu* les moins élevés. Ce phénomène s'explique principalement par le faible nombre d'observations, dans ce cas le vrai graphe et le graphe optimal en terme de score différent. Nous avons déjà pu observer ce phénomène sur la Figure **3.9** où SGS^3 atteint un score supérieur à celui du vrai graphe. Dans cette situation, optimiser la fonction de score ne permet pas de réduire systématiquement la distance d'édition au vrai graphe. SGS^3 arrive second avec des distances d'édition proches de celles obtenues par GES lorsque le nombre d'observations est supérieur à 50, hormis pour le réseau Pigs où GES retrouve la vraie structure avec 5 000 observations. SGS^1 et SGS^2 obtiennent des résultats très similaires tandis qu'aucune règle ne ressort afin de les distinguer réellement de LAGD. Avec 50 observations les réseaux appris par toutes les méthodes se dégradent fortement ce qui rend toute interprétation difficile.

Le nombre de FP obtenus par SGS pour le réseau Pigs s'explique par la présence de nombreuses *v-structures couvertes* dues à un mauvais choix dans l'orientation des arcs. Prenons l'exemple de la Figure **3.10** où le vrai graphe G_{vrai} est composé de la *v-structure* $X_1 \rightarrow X_3 \leftarrow X_2$, supposons maintenant que l'algorithme SGS choisisse d'ajouter dans un premier temps un arc entre X_1 et X_3. Sachant que les deux orientations mènent à deux graphes Markov-équivalents, l'une d'entre elles est donc choisie aléatoirement. Prenons le cas où l'arc est orienté vers X_1 (G_1). Puis un arc entre X_2 et X_3 est ajouté, de la même manière les 2 orientations sont Markov-équivalentes, dans notre exemple l'orientation choisie pointe vers X_2 (G_2). Dans cette situation, afin de couvrir la dépendance ($X_1 \not\perp X_2|X_3$) un $3^{\text{ème}}$ arc est ajouté, $X_1 \rightarrow X_2$ ou $X_2 \rightarrow X_1$ menant au graphe final G_3, cet arc, absent du vrai graphe augmente ainsi le nombre de FP. Lorsque la *v-structure* est entièrement indépendante du reste du graphe, la probabilité en espérance de couvrir cette *v-structure* est de 0.75 (3 orientations de la structure $X_1 - X_3 - X_2$ mènent à cette situation sur les 4 possibles). Cette situation est donc très fréquente lorsque le vrai graphe comporte de nombreuses *v-structures*.

Le réseau Pigs représente un pédigré d'une population de porcs et contient de fait un grand nombre de *v-structures* indépendantes (296 exactement), où chacune d'elles de type $X_1 \rightarrow X_3 \leftarrow X_2$ indique que l'individu X_3 est issu de la reproduction des individus X_1 et X_2. De plus nous remarquons que les erreurs structurelles pour SGS sont exclusivement dues à des arcs superflus dès lors que le nombre d'observations est au moins égal à 500. Cette observation semble ainsi indiquer que seules des *v-structures couvertes* sont responsables de ces arcs superflus et qu'il est par conséquent possible de corriger *a posteriori* ce phénomène.

Tableau 3.6 – Nombre de faux positifs (FP) et de faux négatifs (FN) pour chaque méthode suivant le réseau et le nombre d'observations. En gras est indiqué pour chaque situation les valeurs qui induisent la distance d'édition la plus faible (FP+FN).

		Alarm			Insurance		
Nombre d'observations		50	500	5k	50	500	5k
SGS[1]	FP	44	10	10	18	5	3
	FN	14	4	2	32	20	10
SGS[2]	FP	44	10	10	18	5	3
	FN	14	4	2	32	20	9
SGS[3]	FP	44	**8**	6	19	**4**	**1**
	FN	14	**3**	2	32	**20**	**8**
LAGD	FP	31	11	8	15	**4**	5
	FN	14	4	2	32	**20**	11
GES	FP	**18**	**6**	**4**	**12**	2	3
	FN	**19**	**5**	**2**	**34**	23	12
		Hailfinder			Pigs		
SGS[1]	FP	18	17	16	558	36	49
	FN	43	24	14	281	0	0
SGS[2]	FP	18	17	16	560	34	49
	FN	43	24	13	284	0	0
SGS[3]	FP	18	17	**16**	587	32	41
	FN	43	24	**13**	281	0	0
LAGD	FP	17	21	20	n/a	n/a	n/a
	FN	42	26	19	n/a	n/a	n/a
GES	FP	**15**	**15**	11	**121**	**2**	**0**
	FN	**43**	**24**	22	**420**	**7**	**0**

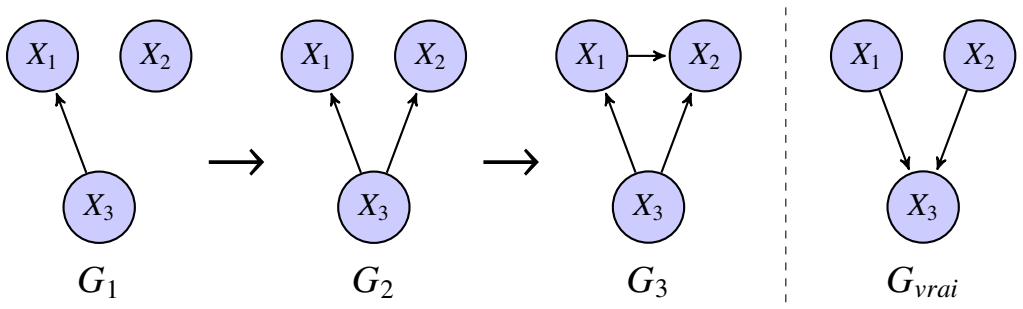


Figure 3.10 – Couverture de la vraie *v-structure* G_{vrai}, par une mauvaise orientation des arcs $X_3 \rightarrow X_1$ (G_1) puis $X_3 \rightarrow X_2$ (G_2), menant à G_3 .

Nous proposons une procédure d'analyse locale des *v-structures* couvertes détectées dans le graphe appris dont le code est présenté dans l'algorithme 3. Cette technique traite chaque *v-structure* couverte indépendamment du reste du graphe, pour cela et afin d'éviter des interdépendances entre *v-structures*, chaque arc composant une *v-structure* détectée est immédiatement retiré du graphe avant de poursuivre la détection. Ainsi l'ensemble des *v-structures* détectées ne possède aucun arc commun, évitant les interdépendances. De fait, cette sélection dépend de l'ordre de parcours des arcs, toutefois dans notre cas, dû au faible nombre d'interdépendances observées, l'impact de cette variabilité n'a pas encore été étudié. On compare alors pour chaque *v-structure* détectée, le score local de chacune des 2 autres *v-structures* non couvertes possibles (dans l'exemple précédent $X_2 \rightarrow X_1 \leftarrow X_3$ et $X_1 \rightarrow X_3 \leftarrow X_2$). La 3ème *v-structure* non couverte ($X_1 \rightarrow X_2 \leftarrow X_3$) est présente dans le voisinage de G_3, celle-ci n'est donc pas testée vu que son score est nécessairement plus faible que celui du graphe final G_3. Si l'une des 2 *v-structures* non couvertes améliore le score par rapport à la *v-structure* couverte alors le graphe courant est modifié. Cette procédure n'assure pas du caractère acyclique du graphe final ni de son optimalité en terme de score global, il est cependant possible d'améliorer cette technique afin de prendre en compte ces deux aspects. Nous pouvons voir sur la Figure **3.11** que ce post-traitement appliqué à un sous-ensemble du réseau Pigs appris lors d'une exécution de l'algorithme SGS^3 permet de retrouver toutes les vraies *v-structures*.

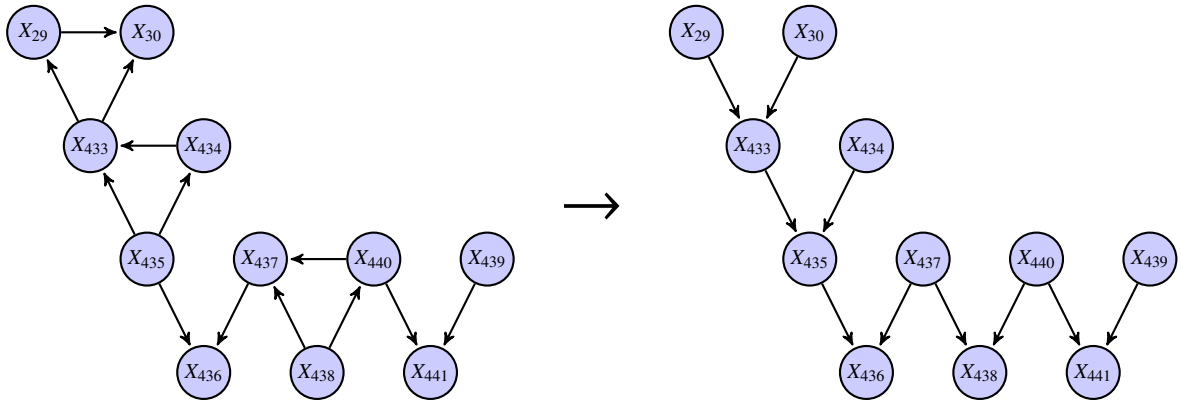


Figure 3.11 – Correction des *v-structures couvertes* appliquée à un sous ensemble du réseau Pigs appris (à gauche).

Certains Faux Positifs persistent cependant comme nous pouvons l'observer sur le Tableau **3.7** qui indique les nouvelles valeurs de distances d'édition pour l'algorithme SGS sur les 4 réseaux. Une baisse du nombre de FP est visible sur quasiment tous les réseaux hormis pour Insurance. Le résultat le plus marquant se situe naturellement pour le réseau Pigs dû au nombre élevé de *v-structures*. On note également que le nombre de FN reste stable indiquant qu'aucun arc correct n'est enlevé à tort par le post-traitement ce qui permet à SGS^3 d'obtenir la meilleure distance d'édition dans la majorité des configurations testées.

Algorithm 3: Procédure de correction des *v-structures couvertes*.

Input : observations D, score f, DAG courant $G(\mathbf{X}, \mathbf{E})$
Output : graphe modifié $G(\mathbf{X}, \mathbf{E})$
$\mathbf{L} \leftarrow \emptyset$ /* *Initialisation à l'ensemble vide de la liste des v-structures couvertes*/ ;
for $(X \rightarrow Z) \in \boldsymbol{E}$ **do**
  /* *Parcours chaque arc du graphe* */ ;
  if $\exists((Y \rightarrow Z) \in \boldsymbol{E} \wedge (X \rightarrow Y) \in \boldsymbol{E})$ **then**
    /* *v-structure* $X \rightarrow Z \leftarrow Y$ *couverte par* $X \rightarrow Y$ */;
    $\mathbf{L} \leftarrow triplet(X, Z, Y)$;
    $\mathbf{E} = \mathbf{E} \setminus \{(X \rightarrow Z), (Y \rightarrow Z), (X \rightarrow Y)\}$;
for $triplet(X, Z, Y) \in \boldsymbol{L}$ **do**
  /* *Calcul du score de la v-structure couverte* */ ;
  $score_c \leftarrow \Delta_Z(\emptyset, ADD(X \rightarrow Z, Y \rightarrow Z) + \Delta_Y(\emptyset, ADD(X \rightarrow Y))$;
  /* *Calcul du score des v-structures non couvertes* */ ;
  $score_X \leftarrow \Delta_X(\emptyset, ADD(Y \rightarrow X, Z \rightarrow X)$;
  $score_Y \leftarrow \Delta_Y(\emptyset, ADD(X \rightarrow Y, Z \rightarrow Y)$;
  /* *Sélection de la meilleure configuration* */ ;
  if $score_c > max(score_X, score_Y)$ **then**
    $\mathbf{E} = \mathbf{E} \cup \{(X \rightarrow Z), (Y \rightarrow Z), (X \rightarrow Y)\}$;
  else
    if $score_X > score_Y$ **then**
      $\mathbf{E} = \mathbf{E} \cup \{(Y \rightarrow X), (Z \rightarrow X)\}$;
    else
      $\mathbf{E} = \mathbf{E} \cup \{(X \rightarrow Y), (Z \rightarrow Y)\}$;
return G ;

Tableau 3.7 – Nombre de faux positifs (FP) et de faux négatifs (FN) pour chaque méthode suivant le réseau et le nombre d'observations après correction des *v-structures couvertes*. Les résultats de GES sont rappelés pour comparaison.

		Alarm			Insurance		
Nombre d'observations		50	500	5k	50	500	5k
SGS[1]	FP	39	8	5	17	5	3
	FN	15	4	2	32	21	10
SGS[2]	FP	39	7	5	17	4	3
	FN	15	4	2	32	20	9
SGS[3]	FP	40	**6**	**2**	17	**4**	**1**
	FN	14	**3**	**2**	32	**20**	**8**
GES	FP	**18**	6	4	**12**	2	3
	FN	**19**	5	2	**34**	23	12
		Hailfinder			Pigs		
SGS[1]	FP	15	16	13	546	3	7
	FN	43	24	14	281	0	0
SGS[2]	FP	16	16	13	548	3	7
	FN	43	24	14	284	0	0
SGS[3]	FP	**15**	16	**13**	574	**1**	2
	FN	**43**	24	**13**	281	**0**	0
GES	FP	**15**	**15**	11	**121**	2	**0**
	FN	**43**	**24**	22	**420**	7	**0**

3.6 Conclusion

Nous avons présenté dans ce chapitre l'algorithme stochastique SGS qui parcourt l'espace des DAG en effectuant à chaque ajout d'arc un choix aléatoire sur l'orientation de celui-ci, si les deux graphes résultants sont équivalents en terme de score. Cet algorithme permet, sous condition d'être répété plusieurs fois, de simuler un déplacement dans l'espace des cpDAG. Deux nouveaux opérateurs ont également été introduits. En premier l'opérateur SWAP permet d'effectuer une suppression et un ajout d'arc durant la même itération afin d'échapper à des optima locaux. Dans un deuxième temps une extension des opérateurs disponibles a été proposée sous le nom d'opérateurs itératifs afin de contourner la contrainte d'acyclicité. La recherche sort alors temporairement de l'espace des DAG en autorisant l'application d'opérateurs créant un circuit sous condition de trouver un ensemble d'opérations associées permettant de restaurer l'acyclicité du graphe. L'ensemble des opérations ainsi appliquées permet d'améliorer le score global du graphe et d'éviter à nouveau certaines situations d'optima locaux.

L'algorithme SGS accompagné de ces nouveaux opérateurs a permis d'améliorer le score *BDeu* atteint sur 4 réseaux benchmarks à partir d'ensembles d'observations de tailles variées face à deux

autres algorithmes performants. L'amélioration la plus importante est observée lors de l'utilisation des opérateurs itératifs qui ont en outre montré qu'une initialisation par un graphe aléatoire ne dégrade pas obligatoirement la qualité du réseau appris. Par ailleurs le calcul des distances d'édition obtenues par chaque méthode a mis en lumière le fait qu'améliorer la fonction de score ne se traduit pas nécessairement par une réduction de la distance d'édition au vrai graphe. Ce constat étant principalement lié au faible nombre d'observations utilisées pour construire nos jeux de données. Cette distance d'édition peut toute fois être améliorée *a posteriori* pour certaines structures présentant une structure typique permettant ainsi de réduire le nombre de *Faux Positifs*.

Par faute de temps, nous n'avons pas pu comparer de manière approfondie ces propositions dans le cadre des données de *génétique-génomique*. Nous montrerons tout de même en section 7.2.2.6 une application de l'algorithme SGS et des nouveaux opérateurs sur des données génétiques, cependant celles-ci ne comportent que des niveaux d'expression de gène sans inclure les données de mutations.

Chapitre 4

Propositions d'évolution à chaque étape de l'apprentissage

Sommaire

4.1 Nouvelles politiques de discrétisations 86

4.1.1 Discrétisations univariées 86

4.1.2 Discrétisation multivariée itérative 87

4.2 Filtre adapté au score 91

4.3 Proposition de scores étendus 92

4.4 Conclusion 94

Dans ce chapitre nous présentons quelques contributions apportées aux différentes étapes de l'apprentissage d'un réseau bayésien. La première d'entre elles porte sur la discrétisation des données où nous proposons deux nouvelles stratégies de discrétisation. La première consiste en une discrétisation adaptée au cas des données biologiques tandis que la seconde plus générique permet de fournir un ensemble de discrétisations pour chaque variable guidé par l'information mutuelle. Puis nous présentons une alternative aux couvertures de Markov afin de réduire l'espace de recherche *a priori* à l'aide d'un filtre utilisant la même fonction de score que celui employé pour apprendre la structure. Nous développons également une extension possible des fonctions de score classiques au travers du concept de scores étendus afin de restaurer une probabilité uniforme sur les classes d'équivalence plutôt que sur l'ensemble des structures.

Hormis cette dernière évolution portant sur les fonctions de score, les autres propositions répondent à des besoins opérationnels pour l'analyse de données biologiques continues et présentant un nombre élevé de variables. De fait, ces propositions n'ont pas été étudiées de manière exhaustive d'un point de vue théorique et expérimental par rapport aux autres approches proposées dans la littérature. Une validation plus complète de ces propositions reste donc à effectuer.

4.1 Nouvelles politiques de discrétisations

Nous avons parcouru en section 2.2 différentes approches de discrétisation notamment des techniques dites *supervisées* qui définissent leur politique de discrétisation en fonction des observations. Parmi ces techniques nous avons présenté des approches univariées basées sur la recherche de modèles de mélange ainsi qu'une méthode multivariée mesurant l'information mutuelle entre paires de variables. Nous proposons dans les deux sections suivantes une nouvelle stratégie pour chacune des deux approches.

4.1.1 Discrétisations univariées

Les techniques de discrétisation univariées discrétisent chaque variable de manière indépendante, parmi celles-ci la méthode des K-means permet d'effectuer une discrétisation pour un nombre de classes K fixé. Dans le cas de données d'expression de gènes le paramètre K est généralement fixé à 3 afin de dissocier le niveau d'expression normal d'une sous et d'une sur-expression du gène. De plus ces deux états extrêmes revêtent souvent en biologie un caractère rare. Ce sens ainsi donné à ces 3 classes permet d'imposer une nouvelle contrainte portant sur les effectifs de chacune d'elles.

Discrétisation par K-means adaptée aux données d'expression Nous présentons une version modifiée des K-means adaptée aux cas des données d'expression de gènes prenant en compte le caractère rare des deux niveaux extrêmes d'expression. L'algorithme modifié utilise un nombre de classes initial K_{init} supérieur au nombre de classes attendu K_{obj}, typiquement $K_{init} = 5$ et $K_{obj} = 3$. Puis l'algorithme tente de fusionner les classes extrêmes contiguës afin d'assurer que ces regroupements ne comportent pas moins de 5% des observations sous la contrainte forte de ne pas dépasser 30%. Cette contrainte forte assure le caractère rare de ces situations. Une fois ces deux classes extrêmes formées, les classes intermédiaires restantes sont à leur tour fusionnées afin d'obtenir les 3 classes désirées. Cependant l'existence d'un tel regroupement en 3 classes n'est pas assurée, prenons ainsi l'exemple d'une discrétisation par K-means classique répartissant les observations en 5 classes dont les effectifs sont tels que $[37\%, 17\%, 28\%, 14\%, 4\%]$. Les premiers regroupements visent à former la classe sous-exprimée, malheureusement la première des 5 classes contient déjà plus de 30% des observations, celle-ci ne représentant donc pas un caractère rare, aucune classe sous-exprimée n'est alors définie. Puis la classe sur-exprimée est obtenue par fusion des deux dernières classes permettant ainsi de représenter plus de 5% des observations sans en dépasser 30%. Enfin la fusion des classes intermédiaires forme alors une discrétisation binaire d'effectifs $[82\%, 18\%]$. Cette méthode de discrétisation est particulièrement adaptée lorsque la distribution des observations pour une variable est approximativement gaussienne comme c'est le cas de la variable $C082$ présentée sur la Figure **4.1**. On observe qu'une discrétisation par recherche de modèles de mélange ne détecte qu'un seul mode tandis que notre discrétisation par K-means adaptée apparaît plus naturelle que celle obtenue par l'algorithme des K-means classique.

Si l'on regarde maintenant les discrétisations effectuées pour la variable $C017$, typique d'une distribution de mélange, la recherche de modèle de mélange obtient logiquement une discrétisation adaptée en comparaison des deux approches de type K-means qui cherchent à définir les 3 classes

désirées. Il n'existe donc pas de méthode unique permettant de discrétiser de manière satisfaisante les différentes distributions possibles, il est nécessaire d'adapter le choix de la technique de discrétisation à la distribution sous-jacente des observations. Nous présentons ainsi une approche adaptative dans ce but.

Discrétisation adaptative Notre approche adaptative consiste à détecter dans un premier temps le nombre K_{pic} de modes visibles d'après les observations. Cette détection s'effectue par un lissage de l'histogramme des observations de la variable cible puis d'une technique de détection des pics significatifs. Lorsque $K_{pic} > 1$, une recherche de modèle de mélanges est utilisée afin de déterminer une discrétisation basée sur les K_{pic} distributions. Si $K_{pic} = 1$ la discrétisation s'effectue par la méthode des K-means proposée précédemment. Cette technique ne fournit donc pas un nombre de classes fixe, celui-ci s'adaptant en fonction des observations. Cependant, afin de ne pas créer des variables discrétisées ayant des domaines trop élevés, un nombre maximal K_{max} de classes est fixé. Pour respecter cette borne maximale, les paramètres du lissage de l'histogramme sont adaptés progressivement tant que $K_{pic} > K_{max}$. La discrétisation adaptative permet de choisir la recherche de modèle de mélange pour la variables $C017$ et notre méthode modifiée des K-means pour la variable $C082$ (voir Figure **4.1**).

Bien que le nombre de paramètres nécessaires afin d'effectuer le lissage et la détection des pics est important, ceux-ci peuvent être ajustés automatiquement en fonction du nombre d'observations. Cette discrétisation adaptative repose principalement sur une analyse visuelle des histogrammes qui ne dépend d'aucun critère statistique et qui peut donc être critiquée en ce sens. Cependant cette approche facilite l'interprétation de la discrétisation dans le cas de données biologiques.

4.1.2 Discrétisation multivariée itérative

A la différence des méthodes précédentes, les approches multivariées discrétisent chaque variable en fonction de l'état des autres variables. Dans cette optique Hartemink [2001] propose une méthode débutant d'une discrétisation initiale composée d'un nombre de classes K_{init} élevé, et qui effectue successivement pour chaque variable le regroupement de deux classes contiguës afin de minimiser la perte de l'information mutuelle entre cette variable et les autres variables du modèle. Cette méthode se répète jusqu'à obtenir le nombre de classes K_{obj} désiré. D'une manière similaire nous proposons une nouvelle méthode de discrétisation itérative guidée par l'information mutuelle et nécessitant une discrétisation initiale. Cependant à la différence de Hartemink [2001] les informations mutuelles sont calculées sur un ensemble restreint de variables (appelés *référentes*), de plus notre approche génère un ensemble de politiques de discrétisation. L'algorithme 4 décrit le fonctionnement général de cette approche.

La première étape de la discrétisation consiste à calculer l'information mutuelle de toutes les paires de variables sachant une discrétisation initiale des observations dont le nombre de classes K_{init}

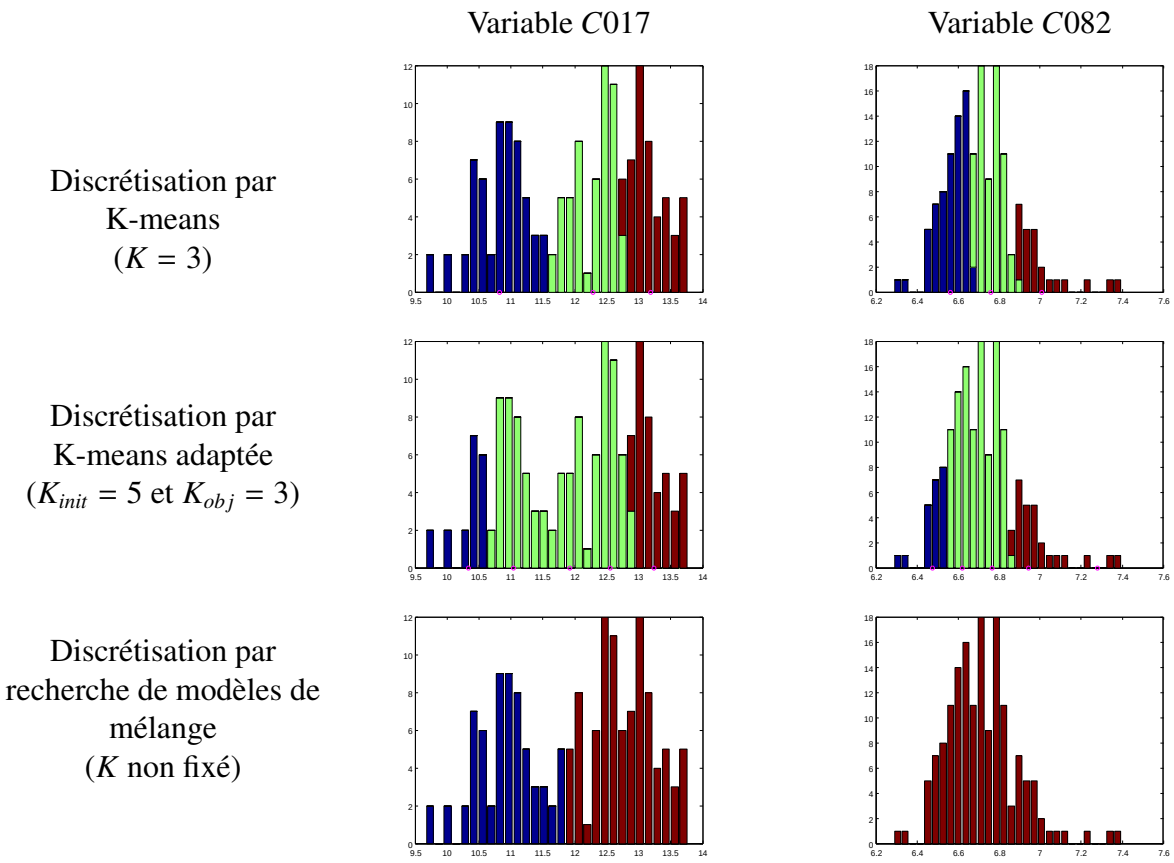


Figure 4.1 – Comparatif des discrétisations effectuées par les méthodes de K-means classique, de K-means adaptée ainsi que de recherche de modèles de mélange pour l'expression de deux gènes issus d'observations réelles sur la plante Arabidopsis thaliana.

est supérieur au nombre final attendu K_{obj} (au maximum K_{init} est égal au nombre d'observations) (ligne 14). Puis on sélectionne pour chaque variable l'ensemble des $nbRef$ autres variables ayant l'information mutuelle la plus élevée avec celle-ci. Ces variables corrélées forment alors l'ensemble des *référents* de la variable considérée (ligne 15). Par la suite l'algorithme itère successivement l'appel à deux fonctions permettant de définir à chaque itération une nouvelle politique de discrétisation pour l'ensemble des variables. La fonction $PerfectMatching$ de la ligne16 établit une liste optimale de paires de variables où chaque variable n'appartient qu'à une seule paire, sélectionnées afin de maximiser la somme des informations mutuelles entre les deux variables composant chaque paire formée. Puis la fonction $OptimiseDiscret$ détermine pour chaque paire, la politique de discrétisation optimale de chacune des deux variables sachant leurs variables *référentes* respectives (ligne 17).

Cette fonction agit par énumération de toutes les discrétisations conjointes possibles de ces deux variables afin de sélectionner la discrétisation qui maximise la somme de l'information mutuelle entre ces deux variables et des informations mutuelles entre chacune d'elles et leurs *référents* respectifs. Lors de ce calcul seules les deux variables du couple sont discrétisées suivant le nombre de classes attendu, tandis que les variables *référentes* gardent la discrétisation fournie en entrée de l'algorithme

afin d'éviter des problèmes d'ordre pour le traitement des couples.

La politique de discrétisation pour ces deux variables est alors mémorisée à la ligne 18 puis l'information mutuelle associée est mise à 0 dans la matrice M afin d'interdire que le couple soit sélectionné à nouveau à l'itération suivante. A chaque itération des couples différents sont alors formés faisant varier de fait les politiques de discrétisation générées. Supposons que (X_1,X_7) représente l'un des couples constitués lors de la première itération. La fonction $OptimiseDiscret$ définit alors une politique de discrétisation pour chacune de ces deux variables maximisant les informations mutuelles conditionnelles liées à X_1 et X_7 (entre elles et par rapport à leurs référents respectifs). Lors de la seconde itération la fonction $PerfectMatching$ associe X_1 à une autre variable (X_7 étant maintenant interdite), supposons X_2. Cette fois donc, la recherche de la politique optimale pour X_1 sera guidée par les informations mutuelles conditionnelles liées à X_1 et X_2. On obtient ainsi au terme de r itérations, un ensemble de r politiques différentes de chaque variable. Du fait que la fonction $PerfectMatching$ forme, à chaque itération, la liste optimale des paires de variables (parmi les couples encore possibles), la somme des informations mutuelles de chaque paire formée diminue au fil des itérations. Bien que la borne supérieure de $nbIterMax$ soit fixée à $(p-1)$, il est donc souhaitable de limiter $nbIterMax < 10$ pour ne pas chercher à maximiser des informations mutuelles entre paires de variables peu corrélées.

Algorithm 4: Procédure de discrétisation itérative.

Input : nombre d'itérations $nbIterMax$, nombre de référents $nbRef$, discrétisation initiale α
Output : ensemble $allPol$ des politiques de discrétisation
$\mathcal{M} \leftarrow zeros(p,p)$ /* Initialise à zéro la matrice des informations mutuelles ($p \times p$) */ ;
$referents \leftarrow zeros(p, nbRef)$ /* Initialise à zéro la matrice des référents ($p \times nbRef$) */ ;
$allPol \leftarrow \emptyset$ /* Initialise à vide l'ensemble des politiques ($nbIterMax \times p$) */ ;
for (i,j) *tel que* $i \neq j$ **do**
    /* Considère tout les couples possibles */ ;
14 $\mathcal{M}(i,j) \leftarrow IM(i,j)$ /* Calcul de l'information mutuelle */ ;
for $i \in [1..p]$ **do**
15 $referents[i] \leftarrow max_{ij}^{nbRef}(\mathcal{M}(i,j))$ /* Retient les $nbRef$ variables les plus corrélées avec i */ ;
for $nbIter = 1 \rightarrow nbIterMax$ **do**
16 $\mathcal{E} \leftarrow PerfectMatching(\mathcal{M})$ /* Sélection de l'ensemble des $\frac{p}{2}$ couples optimaux */ ;
    for $(i,j) \in \mathcal{E}$ **do**
        /* Pour chaque couple sélectionné */ ;
17      $\{bestPol_i, bestPol_j\} \leftarrow OptimiseDiscret(i,j)$ /* Retourne la meilleure politique pour i et j*/ ;
18      $allPol[nbIter][i] \leftarrow bestPol_i$ /* Mémorise les politiques de i et j */ ;
        $allPol[nbIter][j] \leftarrow bestPol_j$;
        $\mathcal{M}(i,j) \leftarrow 0$ /* Mise à 0 de l'IM correspondante */ ;
return $allPol$;

Influence de la discrétisation initiale Le choix du nombre de classes K_{init} de la discrétisation initiale est un paramètre important de l'algorithme. Une valeur suffisamment élevée assure une bonne exploration des discrétisations possibles mais augmente de fait le nombre de discrétisations jointes à

tester $(C_{K_{init}}^{K_{obj}})^2$ et donc la complexité de l'algorithme. Afin de réduire cette complexité nous utilisons un paramètre *step* réglant la valeur du pas lors de l'énumération des politiques de discrétisations. Les bornes de chaque classe sont ainsi déplacées de la valeur *step* le long du domaine de définition des observations afin de passer d'une politique d'énumération à une autre.

Prenons l'exemple d'une discrétisation initiale où chaque classe contient une seule observation, soit $K_{init} = n$. La première politique de discrétisation en K_{obj} classes pour le couple de variables (X, Y) est définie comme suit :

$$\Delta_X = \{-\infty, x_{step}, x_{2step}, \ .. \ , x_{(K_{obj}-1)step}, +\infty\} \ ; \ \Delta_Y = \{-\infty, y_{step}, y_{2step}, \ .. \ , y_{(K_{obj}-1)step}, +\infty\}$$

où x_{step} représente la valeur de la $step^{ième}$ observation triée pour la variable X.

Ces 2 politiques correspondent à la situation où les *step* premières observations sont placées dans la première classe, puis les *step* observations suivantes dans la seconde classe et ainsi de suite jusqu'à la dernière classe qui contient toutes les observations restantes $(n - (K_{obj} - 1)step)$. L'énumération des politiques de discrétisation *voisines* s'effectue en transférant *step* observations entre deux classes contiguës pour une seule des 2 politiques. Ainsi le transfert de *step* observations entre les deux dernières classes de X mène au couple de politiques suivant :

$$\Delta_X = \{-\infty, x_{step}, x_{2step}, \ .. \ , x_{(K_{obj})step}, +\infty\} \ ; \ \Delta_Y = \{-\infty, y_{step}, y_{2step}, \ .. \ , y_{(K_{obj}-1)step}, +\infty\}$$

Augmenter la valeur du paramètre *step* réduit ainsi le nombre de discrétisations possibles et donc la complexité de l'algorithme mais réduit d'autant la précision de la discrétisation obtenue. On remarquera par ailleurs qu'il n'est pas nécessaire de recalculer toutes les informations mutuelles afin de mesurer la qualité d'une politique de discrétisation voisine. Dans notre cas seules les informations mutuelles liées aux deux dernières classes de X vis-à-vis de ses référents doivent être recalculées.

De manière plus générale l'indépendance de chaque paire au sein d'une même itération de l'algorithme permet de paralléliser les discrétisations ce qui diminue d'autant la durée nécessaire sur des machines multi-cœurs.

Comparaison avec la discrétisation par regroupement de Hartemink La première différence provient de l'approche choisie afin de progresser parmi les politiques de discrétisation, alors que notre algorithme énumère directement des discrétisations en K_{obj} classes, Hartemink [2001] préfère une descente gloutonne vers le nombre de classe souhaité. De plus dans notre approche les discrétisations des deux variables de chaque couple sont interdépendantes, tandis que l'approche d'Hartemink [2001] effectue le meilleur regroupement pour une variable donnée indépendamment des regroupements de même niveau pour les autres variables. Ces différences réduisent le nombre de politiques explorées par la technique des regroupements à $p * \frac{K_{init}-K_{obj}-1}{2}$ pour p variables. Notre approche permet quant à elle une recherche plus exhaustive tout en conservant la possibilité de réduire sa complexité à l'aide d'un paramètre dédiée (le *step*).

La deuxième différence provient de l'utilisation dans notre algorithme d'un ensemble de variables *référentes* pertinentes pour chaque variable ainsi que la discrétisation simultanée par paire. Les

politiques de discrétisation générées favorisent ainsi les relations entre les deux variables composant les paires sélectionnées ainsi que celles vis-à-vis des référents. Hartemink [2001] choisi au contraire de guider le choix de la discrétisation par l'ensemble des relations possibles, bien que certaines d'entre elles, voir la majorité dans le cas de réseaux peu denses, ne soient pas pertinentes du fait qu'elles mesurent des corrélations entre des variables éloignées dans le réseau.

L'algorithme proposé bien qu'implémenté n'a pas encore été comparé aux autres techniques de discrétisation.

4.2 Filtre adapté au score

La complexité d'une recherche dans l'espace entier des DAG ou même des cpDAG augmente avec le nombre de variables. Nous avons vu en section 2.1.3 différentes méthodes permettant de restreindre l'espace de recherche afin de traiter des graphes de grande taille. Parmi elles l'algorithme *Sparse Candidate* [Friedman et al., 1999b] sélectionne l'ensemble des k meilleurs parents pour chaque variable suivant la fonction de score. Cependant l'utilisation d'une valeur de k fixée arbitrairement et commune à toutes les variables peut amener à restreindre trop fortement l'espace de recherche et à supprimer des arcs présents dans la vraie structure.

Afin d'adapter la stringence du filtre à la variable considérée nous effectuons un test qui sélectionne pour chaque variable X_i toute variable nécessaire à un ensemble de parents qui améliore le score local à X_i par rapport à l'absence de parents. Cette sélection s'effectue de manière progressive en augmentant à chaque itération la taille l des ensembles de parents considérés jusqu'à une taille maximale fixée l_{max}. Cette borne maximale ne correspond cependant pas au nombre maximum de parents autorisés dans le graphe durant la recherche.

Lors de la première itération seules des situations à un seul parent sont testées, pour chaque variable X_i l'ensemble des parents potentiels $\mathcal{PP}^1_{X_i}$ de niveau 1 est tel que

$$\mathcal{PP}^1_{X_i} = \{\, X_j \mid \Delta_{X_i}(\emptyset, ADD(X_j \rightarrow X_i)) > 0 \,\}$$

où $\Delta_{X_i}(Pa(X_i), Op)$ représente la variation du score local à X_i produite par l'opération Op sachant ses parents actuels ($Pa(X_i)$). Pour notre filtre nous comparons les opérations d'ajout à la situation d'absence de parents pour X_i, soit $Pa(X_i) = \emptyset$.

L'itération suivante teste chaque ensemble composé de 2 parents (X_j, X_k) et sélectionne le couple si celui améliore le score par rapport à la situation où X_i n'a aucun parent mais aussi par rapport aux 2 situations telles que X_i possède X_j ou X_k comme unique parent (testé précédemment). Cette itération définit l'ensemble de parents potentiels $\mathcal{PP}^2_{X_i}$ de niveau 2 tel que

$$\begin{aligned}\mathcal{PP}^2_{X_i} = \{\, X_j, X_k \mid \Delta_{X_i}(\emptyset, ADD(X_j \rightarrow X_i, X_k \rightarrow X_i)) > \\ max(0,\ \Delta_{X_i}(\emptyset, ADD(X_j \rightarrow X_i)),\ \Delta_{X_i}(\emptyset, ADD(X_k \rightarrow X_i)) \,\}\end{aligned} \tag{4.1}$$

Chaque filtre de niveau l permet de détecter des interactions complexes impliquant l parents qui n'auraient pu être détectées à un niveau inférieur.

Finalement l'ensemble des parents potentiels pour X_i est défini par l'union de tous les ensembles constitués jusqu'au niveau l_{max}

$$\mathcal{PP}_{X_i} = \cup_{l=1}^{l_{max}} \mathcal{PP}^l_{X_i}$$

Ce test est effectué une seule fois avant même l'exécution de la recherche et utilise la même fonction de score que celle employée lors de la recherche de la structure. A l'issu de ce filtre chaque variable dispose d'un ensemble de parents potentiels de taille variable cohérent avec la recherche devant s'effectuer par la suite. En pratique le paramètre $l_{max} \leq 2$ dû à l'augmentation exponentielle des combinaisons possibles à l parents.

4.3 Proposition de scores étendus

Cas des modèles linéaires Lors de la définition des fonctions de score en section 2.1.2.1 nous avons supposé que la probabilité *a priori* du graphe $\mathbb{P}(\mathcal{G})$ est uniforme en l'absence de connaissance experte. Cependant cet *a priori* uniforme sur les graphes induit un biais de la probabilité *a priori* sur la connectivité des graphes.

Cette idée découle des travaux de Chen and Chen [2008] portant sur la sélection de modèles linéaires où ceux-ci notent qu'il existe naturellement plus de modèles possible à k variables explicatives parmi p, que de modèles n'en possédant que $k-1$ (tant que $k \leq \frac{p}{2}$). Notons par exemple $\mathcal{M}^k$ l'ensemble des modèles comportant k variables explicatives avec $\tau(\mathcal{M}^k)$ le nombre de modèles composant cet ensemble, ainsi nous avons $\tau(\mathcal{M}^k) = C_p^k$ et $\tau(\mathcal{M}^k) > \tau(\mathcal{M}^{k-1})$ pour $k \leq \frac{p}{2}$. Dans ce cas, la probabilité *a priori* de sélectionner un modèle m comportant k variables explicatives est $\mathbb{P}(m \in \mathcal{M}^k) = \tau(\mathcal{M}^k)\mathbb{P}(m)$ avec $\mathbb{P}(m) = \frac{1}{\sum_{i=0}^{p} \tau(\mathcal{M}^i)}$ la probabilité *a priori* uniforme du modèle. La probabilité $\mathbb{P}(m \in \mathcal{M}^k)$ augmente donc avec la valeur de k.

Cette observation tend à la sélection de modèles plus complexes allant à l'encontre du dogme des réseaux biologiques peu denses. Afin de palier cette incohérence, Chen and Chen [2008] définissent un critère *BIC* étendu permettant d'assurer une probabilité $\mathbb{P}(m \in \mathcal{M}^k)$ uniforme ce qui implique $\mathbb{P}(m) \propto \frac{1}{\tau(\mathcal{M}^k)}$.

Application aux réseaux bayésiens Cette approche se transpose au cas des réseaux bayésiens où la probabilité de sélectionner un graphe augmente avec les degrés entrant des variables qui le compose. Ainsi nous proposons d'utiliser un *a priori* uniforme sur les classes de graphes ayant les mêmes degrés entrants pour chaque variable. Pour ce faire nous définissons la probabilité *a priori* de chaque graphe $\mathcal{G}_i$ par

$$\mathbb{P}(\mathcal{G}_i) \propto \tau(\mathcal{G}_i)^{-\gamma}$$

avec $\gamma \in [0, 1]$, $\mathcal{G}_i$ le graphe restreint à X_i composé des variables $\mathbf{X}' = \{X_i \cup Pa(X_i)\}$ reliées par $\mathbf{E}' = \{\overrightarrow{(X_j, X_i)} \,|\, X_j \in Pa(X_i)\}$ et $\tau(\mathcal{G}_i) = C_{p-1}^{k_i}$ le nombre de combinaisons de $k_i = |Pa(X_i)|$ parents possibles sur les $p-1$ parents potentiels. La valeur $\gamma = 0$ correspond à un *a priori* uniforme sur l'ensemble des DAG, tandis que la valeur $\gamma = 1$ définit un *a priori* uniforme sur les classes de

connectivité.

Afin d'obtenir une probabilité sur le graphe global $\mathbb{P}(\mathcal{G})$, nous supposons dans un premier temps l'indépendance des graphes restreints, $\mathbb{P}(\mathcal{G}) \approx \prod_{i=1}^{p} \mathbb{P}(\mathcal{G}_i)$. Cette hypothèse est cependant remise en cause par les dépendances entres variables dues à la contrainte d'acyclicité des réseaux bayésiens à la différence des modèles linéaires. Ces dépendances biaisent ainsi le calcul du nombre de graphes possibles pour chaque classe de connectivité. Il n'existe par exemple aucun réseau bayésien à deux variables X_i et X_j dont chacune possède un seul parent, tandis que notre approximation du fait d'une énumération locale à chacune des deux variables comptabilisera un réseau cyclique possible où $Pa(X_i) = X_j$ et $Pa(X_j) = X_i$. La Figure **4.2** montre l'écart entre le nombre réel de réseaux bayésiens possibles calculé d'après Favier et al. [2009] pour une distribution des degrés entrants donnée et le nombre estimé par notre approximation pour différentes tailles de réseaux. Cette distribution des degrés sur les variables suit une loi de puissance, avec un nombre d'arcs ≈ nombre de nœuds. On remarque qu'il convient d'utiliser une valeur $\gamma < 1$ ($\gamma = 0.7$) afin de prendre en compte les dépendances entre variables et de rectifier le nombre de graphes possibles dans le but d'obtenir réellement un *a priori* uniforme sur les classes de réseaux bayésiens de même degré entrant.

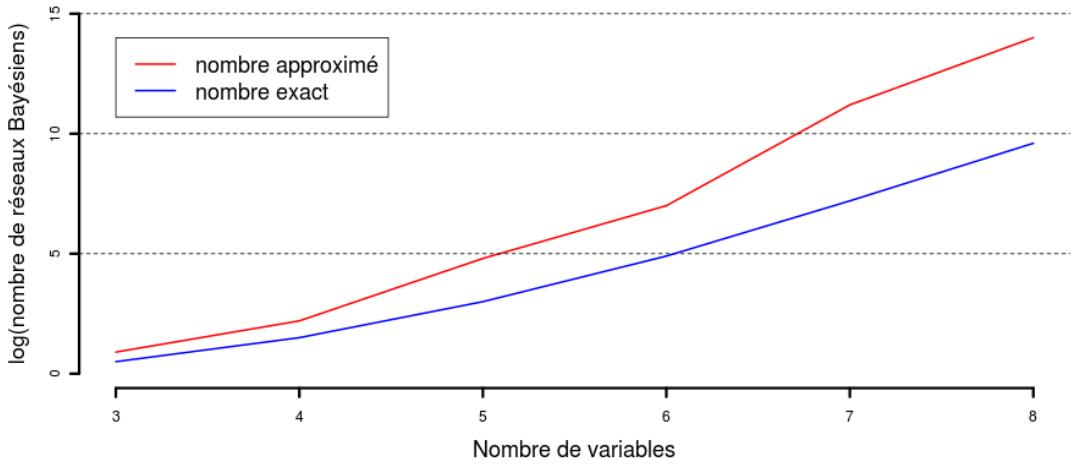


Figure 4.2 – Comparaison du logarithme du nombre réel (en bleu) et approximé (en rouge) de DAG dont la connectivité entrante des nœuds suit une loi de puissance en fonction du nombre de variables du réseau.

L'extension proposée vise donc à fournir un *a priori* non uniforme pour $\mathbb{P}(\mathcal{G})$ et s'applique aux fonctions de score présentées précédemment comme une pondération de leur expression native. On obtient donc pour toute fonction de score S telle que $S(\mathcal{G}) \approx \log \mathbb{P}(\mathbf{D}|\mathcal{G}))$:

$$\begin{aligned} S^{\gamma}(\mathcal{G}) &= \log(\mathbb{P}(\mathbf{D}|\mathcal{G})\mathbb{P}(\mathcal{G})) \\ &= S(\mathcal{G}) + \log(\mathbb{P}(\mathcal{G})) \\ &= S(\mathcal{G}) + \sum_{i=1}^{p} -\gamma \log(\tau(\mathcal{G}_i)) \end{aligned}$$

De plus si S est décomposable on obtient

$$S^{\gamma}(\mathcal{G}) = \sum_{i=1}^{p} -\gamma \log(\tau(\mathcal{G}_i)) + s(\mathcal{G}_i)$$

L'impact de cette extension sera observé lors de comparaisons effectuées à partir de données simulées dans le cadre des réseaux de régulation de gènes en section 7.2.1.

4.4 Conclusion

L'apprentissage de la structure d'un RB ne dépend pas uniquement de la qualité de l'heuristique guidant la recherche dans l'espace considéré (DAG, cpDAG, ordres...) mais d'une succession d'étapes toutes aussi importantes. Ce chapitre présentait ainsi des propositions d'évolution pour trois de ces étapes. La première d'entre elles touche à la discrétisation des données pour laquelle nous avons présenté deux nouvelles approches. La première étant une discrétisation univariée adaptative dédiée aux données biologiques tandis que la seconde, plus générique et multivariée, fournit un ensemble de discrétisations.

La seconde proposition s'applique lorsque les graphes étudiés contiennent un nombre de variables important (typiquement plusieurs milliers) comme cela est généralement le cas des réseaux biologiques. Dans ces situations nous proposons d'utiliser la même fonction de score que celle employée durant la recherche afin de filtrer *a priori* les arcs n'améliorant pas ce score par rapport au graphe vide. Ce filtre volontairement lâche vise à ne pas dégrader trop fortement la recherche par une réduction trop drastique de son espace.

La dernière voie d'évolution explorée concerne la fonction de score dont une extension a été proposée afin d'établir un *a priori* uniforme sur les classes de connectivité des graphes plutôt que sur les graphes eux-mêmes. Cette extension vise en pratique à pondérer davantage les graphes peu denses, afin de prendre en compte l'un des dogmes actuels sur la faible densité des réseaux de régulation de gènes.

La majorité de ces propositions ont été développées dans une optique d'application au problème biologique auquel nous nous sommes intéressé dans cette thèse. Elles seront donc testées dans la partie II de ce manuscrit portant sur la reconstruction de réseaux de régulation de gènes.

Conclusion et perspectives à l'apprentissage de structure

Nous nous sommes attachés dans cette partie à l'apprentissage de réseaux bayésiens génériques et avons vu à cette occasion que la tâche n'est pas aisée en dépit des nombreuses approches proposées. Nous nous sommes concentrés durant cette thèse sur l'apprentissage de réseaux statiques à l'aide d'algorithmes de recherche locale utilisant une fonction de score. Dans ce contexte nous avons proposé deux nouveaux opérateurs ainsi qu'une version stochastique de l'algorithme Greedy Search. Bien que ces propositions se soient révélées efficaces afin d'améliorer la fonction de score employée, de nombreuses perspectives à ce travail restent ouvertes.

La première d'entre elles consiste à compléter notre comparaison avec d'autres méthodes, notamment celles définissant des voisinages locaux étendus tels que nous l'avons proposé, nous pensons ainsi aux travaux de Campos et al. [2002], Moore and Wong [2003]. Il sera alors intéressant de comparer plus en détail les configurations atteignables par chaque voisinage.

De même, le développement de méthodes exactes permet aujourd'hui de trouver les structures optimales en terme de score pour des réseaux de taille modeste (typiquement le réseau Alarm). La connaissance de ces structures optimales offre alors la possibilité d'analyser les configurations empêchant une recherche heuristique d'y parvenir. De nouveaux opérateurs peuvent alors être imaginés afin de contourner ces difficultés tout en limitant leur complexité.

Une deuxième piste concerne l'algorithme SGS. Dans l'état actuel de son développement les différentes exécutions de l'algorithme sont indépendantes, ainsi deux exécutions peuvent effectuer exactement les mêmes choix d'orientation et donc mener au même réseau. Un travail doit donc être mené afin de conserver une trace de ces choix et de ne pas les considérer une seconde fois. Il est également envisageable de pondérer ces orientations en fonction des scores obtenus suite à ces choix, lors de répétitions antérieures. Il sera ainsi possible de guider le choix d'une orientation suivant la qualité des réseaux précédemment atteints à partir de cette même orientation. Cette problématique est d'autant plus critique que le nombre d'orientations équivalentes augmente rapidement avec le nombre d'arcs possibles et que certains choix ont un impact fort sur la suite de la reconstruction.

L'utilisation d'autres heuristiques que la recherche gloutonne doit par ailleurs être explorée, des heuristiques simples telles que l'utilisation d'une liste Tabu peuvent ainsi être envisagées afin d'améliorer les scores atteints. Le développement d'heuristiques complexes ne nous semble cependant pas nécessaire, l'augmentation du nombre de paramètres allant généralement de paire avec ce type de méthode rend souvent leur utilisation difficile.

Nous avons également noté que l'amélioration du score ne se traduisait pas systématiquement par

une augmentation de la qualité structurelle des réseaux. Bien que cette observation soit fortement liée au nombre d'individus utilisés, il est nécessaire de rechercher des moyens permettant de corriger ce phénomène. Outre la modification de la fonction de score, il est possible de considérer des méthodes hybrides composées d'une première recherche heuristique classique telle que SGS, puis d'un post-traitement de ce réseau à l'aide de tests locaux sur des configurations spécifiques afin de réduire cette distance d'édition. Nous avons déjà proposé une telle technique afin de corriger localement les *v-structures* couvertes, il reste cependant à voir si d'autres configurations similaires peuvent être résolues de la sorte.

De plus lors de cette analyse structurelle nous nous sommes restreints à la comparaison du squelette des réseaux appris. Cependant les réseaux bayésiens possèdent une part de liens fortement orientés, identifiables au travers des cpDAG associés. La comparaison des cpDAG en lieu et place des squelettes permet ainsi d'évaluer la capacité des algorithmes à apprendre des relations causales en plus de la simple structure du graphe. Les travaux de Tsamardinos et al. [2006], Nguyen [2012] portant sur la comparaison de graphes essentiels, proposent des solutions dans ce sens ouvrant ainsi la voie à une analyse approfondie de nos résultats.

Plusieurs propositions secondaires ont également été avancées afin d'améliorer la reconstruction du réseau à ses différentes étapes. Certaines de ces propositions offrent également des perspectives intéressantes. La première d'entre elles concerne la discrétisation des données. Deux méthodes de discrétisation ont ainsi été développées dont une sera utilisée dans la deuxième partie de ce manuscrit. Cependant aucune d'elles n'a pu véritablement faire l'objet de comparaisons formelles face à des méthodes récentes de discrétisation. Un état des lieux approfondi du domaine devra ainsi être réalisé afin d'effectuer ce type de comparaison. Il faudra cependant distinguer les objectifs de ces discrétisations qui peuvent diverger. Ainsi si notre première discrétisation repose sur une interprétation biologique des variables, la seconde reste plus générique. De même, notre discrétisation itérative a pour but de proposer un ensemble de discrétisations, se rapprochant ainsi d'une procédure de *bootstrap*. L'intérêt des ces discrétisations multiples reste donc encore à valider expérimentalement.

Un dernier point concerne cette fois la notion de filtre utilisé afin de traiter des réseaux composés d'un nombre élevé de variables. Ce filtre étant principalement appliqué dans le cadre de l'application biologique, nous discuterons des perspectives associées dans la partie suivante.

Deuxième partie

Application à la génétique-génomique

Introduction

La deuxième partie de ce manuscrit présente l'application biologique ayant motivé les travaux présentées en partie 1 à savoir l'apprentissage de réseaux de régulation de gènes. Nous introduirons dans le chapitre 5 le problème biologique auquel nous nous intéressons à savoir l'apprentissage de la structure d'un réseau de régulation de gènes ainsi que la nature des données utilisées. Puis nous effectuerons dans le chapitre 6 un état de l'art consacré cette-fois aux méthodes de reconstruction de réseaux de régulation de gènes sans se restreindre au seul formalisme des réseaux bayésiens. Nous proposerons dans le chapitre 7 une modélisation de ce problème dans le cadre des réseaux bayésiens. Nous présenterons également différentes expérimentations effectuées sur des données simulées. Enfin le chapitre 8 décrira les premiers résultats obtenus sur des données réelles d'Arabidopsis thaliana disponibles au travers d'une collaboration avec des chercheurs des unités INRA de Versailles et d'Évry.

Chapitre 5

Introduction à la génétique-génomique

Sommaire

5.1 Le génome 101
5.1.1 De la cellule au gène 102
5.1.2 Du gène à la protéine 102
5.1.3 L'expression d'un gène 104
5.2 Les régulations géniques 107
5.2.1 Les différents types de régulation 107
5.2.2 Les différents types de réseaux de régulation 108
5.2.3 Les réseaux de régulation de gènes 110
5.3 Polymorphismes et variations d'expression 111
5.4 Conclusion 112

Ce chapitre introduit les bases de biologie nécessaires à la compréhension du problème de reconstruction d'un réseau de régulation de gène, au départ de l'ADN jusqu'au phénomène de régulation. Nous décrivons également les données utilisées dites de *génétique-génomique*. Les hypothèses nécessaires à l'obtention de telles données nous obligent à nous restreindre aux organismes dont la population peut être *contrôlée*, ainsi nous nous focalisons plus particulièrement dans ce chapitre sur les espèces végétales constituées d'organismes eucaryotes multicellulaires.

5.1 Le génome

Le génome représente l'ensemble du matériel génétique nécessaire au développement de toute espèce vivante de la simple bactérie aux mammifères tels que l'homme en passant par les plantes. Ce patrimoine génétique transmis d'une génération à une autre lors de la reproduction est contenu dans le noyau chez les cellules eucaryotes. Chacune de ces cellules comporte la même information génétique de base portée par les gènes, le rôle d'une partie de ces gènes est la production de protéines qui constituent les acteurs principaux de l'activité cellulaire. Tout les gènes ne sont cependant pas actifs

au sein d'une seule cellule. Le sous-ensemble de gènes exprimés définit les fonctions spécifiques de la cellule, menant ainsi à la différenciation cellulaire.

5.1.1 De la cellule au gène

Dans le cas des organismes eucaryotes, l'information génétique est contenue dans le noyau de chaque cellule sous forme de chromosomes comme représenté sur la Figure **5.1**. Les chromosomes sont à leur tour constitués d'une longue molécule d'ADN organisée en une succession d'enroulements autour de complexes de protéines spécifiques, les *histones*, chaque enroulement est appelé un *nucléosome*.

L'ADN est le support physique de l'information génétique, il est constitué de deux brins composés à leur tour d'une succession de *nuclétotides* organisés sous la forme caractéristique d'une double hélice. Les *nucléotides* ou *bases* sont au nombre de 4 : l'adénine (A), la thymine (T), la guanine (G) et la cytosine (C). Ces bases sont complémentaires deux à deux, l'adénine avec la thymine et la guanine avec la cytosine. Les deux brins d'ADN sont alors qualifiés de complémentaires. Ainsi pour chaque nucléotide qui compose l'un des deux brins est associé à la même position sur le second brin, le nucléotide complémentaire, relié par une liaison hydrogène. On utilise alors le nombre de *paires de bases* (pb) comme unité de longueur de l'ADN. Typiquement la dimension du génome (une molécule d'ADN) humain est d'environ $3.4\ 10^9$ pb tandis que le plus grand génome actuel connu est celui de la fleur *Paris Japonica* avec $149\ 10^9$ pb.

Les *gènes* correspondent à des portions de cette double hélice, de tailles variables et réparties le long des chromosomes. Le nombre de gènes varie suivant l'organisme indépendamment de la taille du génome allant de quelques centaines à plusieurs dizaines de milliers. Les gènes représentent les *régions codantes* de l'ADN car celles-ci codent pour la production de molécules spécifiques, les régions restantes, qui représentent une partie importante de l'ADN, sont dites *non-codantes*. On notera cependant qu'un gène n'est pas constitué d'une unique région *codante*, des régions *non-codantes* venant s'intercaler par endroits, un gène est donc une succession de régions *codantes*, les *exons* et de sections *non-codantes*, les *introns*. Le rôle des régions *non-codantes*, un temps ignoré, fait l'objet d'études de plus en plus nombreuses révélant des rôles multiples notamment dans le phénomène de régulation entre gènes.

5.1.2 Du gène à la protéine

La majorité des gènes d'un organisme vise à générer des protéines spécifiques. Le rôle des protéines est multiple et essentiel au sein de l'organisme. Au niveau structurel tout d'abord, comme le cas des fibres protéiques qui forment entre autre l'armature des cellules. D'autres protéines jouent le rôle d'enzyme qui catalyse les nombreuses réactions biochimiques au sein des cellules ou encore le rôle de transmetteur d'informations lorsque l'environnement qui entoure la cellule est modifié. D'autres protéines appelées *facteurs de transcription* permettent de réguler l'activité de certains gènes comme nous le verrons dans la section suivante.

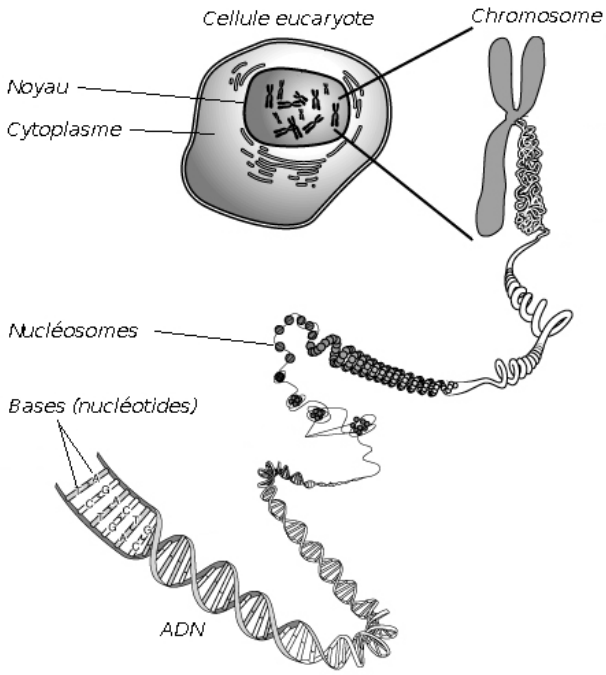


Figure 5.1 – Décomposition du patrimoine génétique pour une cellule eucaryote diploïde.

La Figure **5.2** représente les deux étapes principales nécessaires à la production d'une protéine à partir de l'ADN à savoir une première étape de transcription de la séquence génétique en une molécule d'ARN puis la traduction de cet ARN afin de synthétiser une protéine.

Transcription L'étape de transcription vise à synthétiser une molécule d'ARN constituée d'un seul brin qui soit une copie complémentaire d'un des deux brins de la séquence d'ADN correspondant au gène à transcrire. Pour cela un complexe de protéines se fixe sur une courte région précédent le gène, le *promoteur central*, provoquant ainsi l'*initiation* de la transcription. Puis l'une des enzymes de ce complexe protéique débute sa lecture de la séquence codant le gène à partir du promoteur. Au fur et à mesure de cette progression l'enzyme synthétise la molécule d'ARN, dite *pré-messager* (ARNpm), complémentaire au brin d'ADN parcouru. L'ARNpm est constitué des même bases que celles de l'ADN à l'exception de la thymine, remplacée par l'uridine (U). Ce processus s'arrête lorsque l'enzyme atteint une séquence de quelques nucléotides indiquant la fin de la transcription. L'enzyme ainsi que l'ARNpm se détachent alors du brin d'ADN, marquant la *terminaison* de l'étape de transcription.

Au terme de la transcription l'ARNpm n'est pas encore fonctionnel, différentes modifications vont ainsi lui être appliqué. Parmi ces modifications *post-transcriptionnelles* la phase d'*épissage* consiste à

exciser les *introns* de l'ARNpm qui ont été transcrits afin de conserver uniquement les régions codantes correspondantes au gène. Cet épissage n'est cependant pas déterministe, certains *introns* peuvent ainsi être conservés tandis que des *exons* seront à l'inverse retirés, on parle alors d'*épissage alternatif*. Ces variations ponctuelles génèrent des séquences d'ARNpm différentes à partir d'un même gène menant à terme à une modification des protéines synthétisées. De manière générale il existe au sein d'un organisme, une plus grande variété de protéines que de gènes transcrits.

Une autre des modifications de cet ARNpm vise cette fois à ajouter deux composés aux deux extrémités du brin. Une *coiffe* est tout d'abord ajoutée au début de l'ARNpm, essentiel à la phase de traduction future, puis l'ajout d'une *queue* à la fin de la séquence permet d'allonger sa durée de vie.

A la suite de ces modifications l'ARNpm est dit *mature*, on parle alors d'ARN *messager* (ARNm) prêt à être traduit en protéine.

Traduction A la différence de l'étape de *transcription* se déroulant dans le noyau, la phase de *traduction* s'effectue dans le cytoplasme de la cellule. On retrouve néanmoins une similarité au niveau des étapes entre les deux processus. L'initiation de la *traduction* s'effectue par l'attache d'un complexe au niveau de la *coiffe* de l'ARNm puis l'un des éléments de ce complexe, le *ribosome*, débute sa lecture du brin d'ARNm. La *traduction* s'effectue cette fois par groupe de 3 *nucléotides*, les *codons*. Chacun de ces *codons* indique au *ribosome* l'action spécifique associée, pour la majorité d'entre eux il s'agit de synthétiser des molécules spécifiques, les *acides aminés*, tandis que d'autres servent de ponctuation marquant le début et l'arrêt de la *traduction*. Cette phase dite d'*élongation*, s'arrête lorsque le *ribosome* rencontre un des *codons* d'arrêt de la *traduction*, libérant ainsi la chaîne d'*acides aminés* synthétisée.

Cette chaîne une fois libérée subit à nouveau des modifications *post-traductionelles* afin de lui attribuer une fonction et une position spécifique dans la cellule. Sa structure est modifiée par l'ajout et la suppression de certains acides aminés. A l'issue de ces modifications les protéines sont dites matures et fonctionnelles. On notera par ailleurs que les protéines sont sujettes à ces modifications tout au long de leur vie dans la cellule et que par conséquent un même brin d'ARNm pourra à nouveau synthétiser différentes protéines.

Si la majorité des gènes a pour finalité la synthèse d'une protéine, certains d'entre eux ne subissent pas l'intégralité du processus Transcription-Traduction. Ainsi certains gènes ne sont jamais transcrits tandis que d'autres bien que transcrits en ARN ne sont jamais traduits. Ces ARNs non traduits ne sont pour autant pas dépourvus d'utilité, certains d'entre eux participent même activement aux phénomènes de régulation tels que les micro-ARN qui se fixent à des ARNm spécifiques empêchant ou limitant ainsi leur traduction. D'autres ARNs servent à la construction de composés protéiques, les ARNs ribosomiques (ARNr) entrent ainsi dans la composition des *ribosomes* essentiels à la phase de traduction.

5.1.3 L'expression d'un gène

Protéome vs Transcriptome Le niveau de protéines synthétisées à partir d'un gène n'est pas constant dans le temps, des variations sont observées dû notamment à des phénomènes de régulations.

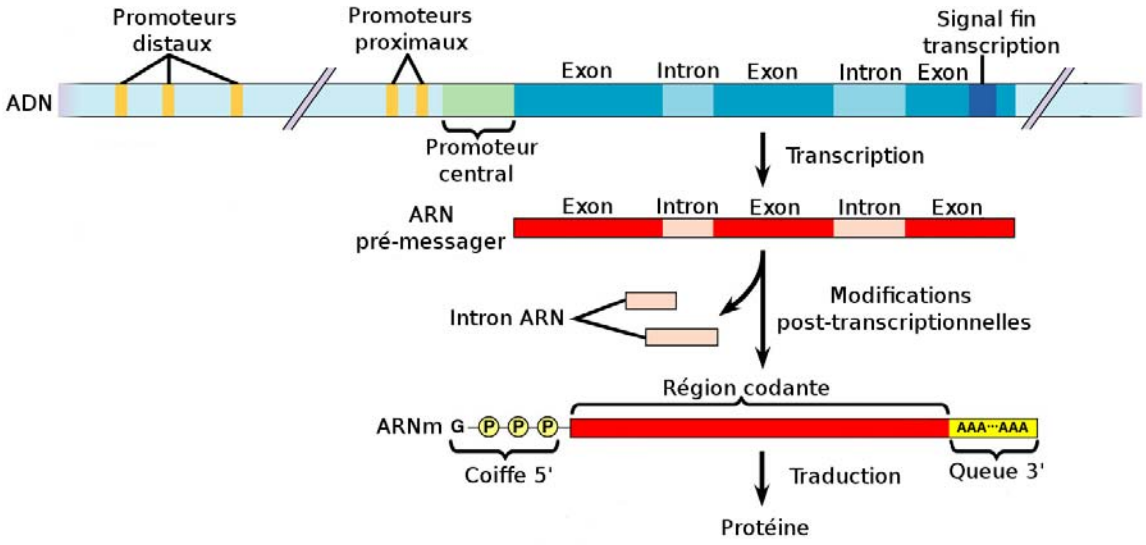


Figure 5.2 – Les différentes étapes nécessaires à la production d'une protéine à partir d'une séquence d'ADN.

Avant de chercher à définir ces régulations il est tout d'abord nécessaire de préciser ce qui caractérise au mieux le niveau d'expression d'un gène. Nous avons vu que chaque gène produit dans le cas général deux types de molécule, les ARNm, issus de la transcription et les protéines lors de la traduction. Il n'y a cependant pas d'équivalence du type *1 gène = 1 ARN = 1 protéine* dû aux modifications post-transcriptionnelles et post-traductionnelles. L'expression d'un gène peut alors être caractérisé de deux manières, soit en analysant le *protéome*, c'est-à-dire l'ensemble des protéines synthétisées, ou bien le *transcriptome* qui correspond à l'ensemble des ARNs transcrits.

Ces deux niveaux d'analyse ne sont pas équivalents dans ce qu'ils permettent d'observer. Ainsi l'analyse des transcrits n'assure nullement que l'ARN soit réellement traduit par la suite en une protéine ni que cette protéine soit unique. L'observation de ces ARNs non traduits reste cependant utile du fait du rôle de certains d'entre eux dans les régulations comme les micro-ARNs. L'analyse du protéome est quant à elle plus précise pour différencier les protéines présentes dans la cellule mais ne permet pas de mesurer les ARNs non traduits. D'un point de vue pratique le nombre plus important de protéines différentes pour un organisme complexifie à la fois l'extraction et le traitement des observations issues du protéome.

Les analyses du protéome et du transcriptome constituent donc des approches complémentaires permettant de représenter le fonctionnement de la cellule de manière complète mais augmente en contre partie la complexité du traitement de ces deux sources de données. Dans cette thèse nous nous sommes restreint au niveau du transcriptome ce qui permet de conserver l'information des ARNs non traduits dont certains jouent un rôle important dans les phénomènes de régulation. Par la suite nous caractériserons l'expression d'un gène uniquement par son niveau de transcrits. Nous noterons que dans la situation d'épissage alternatif, différentes versions du gène peuvent être définies afin de représenter chaque ARN distinct produit.

Analyse du transcriptome Les technologies permettant de mesurer le niveau de transcrits des gènes d'un organisme peuvent être classées en deux catégories : les puces à ADN et le séquençage des ARNm. Nous décrierons brièvement les techniques employées par ces deux approches.

Puces à ADN Les puces à ADN représentent la technique la plus utilisée afin de mesurer l'expression des gènes dû principalement à son faible coût et à la possibilité de mesurer simultanément le niveau de transcrits de milliers de gènes. Concrètement une puce est constituée d'une lame solide sur laquelle sont fixés des fragments d'ADN mono-brin correspondant aux gènes dont on souhaite mesurer l'expression. Ces fragments, les *sondes*, sont regroupés aux sein de *spot*. Chaque *spot* contient plusieurs milliers de *sondes* correspondant au même fragment d'ADN et donc à un gène précis. Ces *spots* sont répartis sur le support de la puce sous la forme d'une grille. Le choix des fragments à inclure sur la puce est primordial du fait que seuls les ARNs correspondants aux *spots* présents seront mesurés.

Les échantillons cellulaires analysés sont typiquement composés d'un ensemble de cellules d'un même tissu afin d'assurer que les mêmes gènes soient exprimés et régulés de façon similaire. Avant d'analyser cet échantillon grâce à la puce, celui-ci doit être préparé. Dans un premier temps les molécules d'ARN sont extraites des cellules puis transformées en séquences d'ADN mono-brin complémentaires (ADNc) lors d'une phase de transcription inverse. Les molécules d'ADNc sont alors marquées à l'aide d'un composé fluorescent ou radioactif afin de quantifier par la suite les niveaux de transcrits, puis l'échantillon est déposé sur la puce. Les ADNc présents dans l'échantillon vont alors s'hybrider naturellement aux sondes correspondant à leur séquence complémentaire. La durée nécessaire à l'hybridation des ADNc dépend de l'organisme étudié et varie typiquement entre 12 et 24 heures. Au bout de ce laps de temps la puce est lavée afin d'enlever les ADNc non hybridés puis l'intensité de chaque spot est mesurée. Cette intensité lumineuse ou radioactive en fonction du type de marquage utilisé, est proportionnelle au nombre d'ADNc hybridés sur chaque spot et donc au niveau d'expression du gène associé.

Il est également possible de mesurer deux échantillons distincts sur une même puce, dite *bicolore*. Dans ce cas chacun des échantillons est marqué d'un composé fluorescent différent, avant d'être mélangé puis déposé sur la puce. Les intensités mesurées permettent une analyse différentielle des niveaux de transcrit entre les 2 échantillons. Les résultats ne sont donc pas équivalents à l'utilisation de deux puces simples différentes et ne permettent pas une comparaison directe à d'autres puces *bicolores*. Ces puces peuvent tout de même être utilisées afin de comparer plus de deux échantillons, pour cela un échantillon commun doit être utilisé entre les différentes paires de puces à comparer.

Séquençage des ARNm L'évolution des techniques de séquençage à haut débit permet progressivement de considérer le séquençage des ARNm comme une solution viable financièrement pour les mesures d'expression. Séquencer une molécule d'ADN ou d'ARN consiste à lire sa succession de nucléotides. Citons par exemple la technique SAGE (Serial Analysis of Gene Expression) qui

concatène un fragment spécifique de chaque ARNm présent dans l'échantillon sous forme d'une chaîne, qui est ensuite séquencée. Le niveau d'expression d'un gène est alors défini comme le nombre d'occurrences du fragment correspondant. Une approche similaire est la méthode MPSS (Massively Parallel Signature Sequencing) qui utilise des fragments plus longs que la technique SAGE réduisant ainsi l'ambiguïté lors du séquençage de fragments semblables. Ces fragments sont ensuite fixés sur un ensemble de micro-billes permettant alors un séquençage en parallèle.

L'un des avantages du séquençage de l'ARN réside dans la précision des niveaux d'expression mesurés. Si les puces à ADN fournissent des données qualitatives sur l'intensité des spots dont la mesure est soumise à plusieurs sources d'erreurs (sensibilité et saturation des capteurs, bruit de fond), le séquençage permet une mesure quantitative des fragments par comptage, moins sensible aux erreurs. De plus le séquençage n'est pas soumis à une liste d'ARNs à mesurer comme c'est le cas des puces à ADN. Tout les ARNs présents dans l'échantillon peuvent ainsi être observés ce qui permet notamment de découvrir de nouveaux ARNs et donc de nouveaux gènes.

5.2 Les régulations géniques

5.2.1 Les différents types de régulation

Nous avons déjà mentionné à plusieurs reprises l'existence de régulations modulant l'expression des gènes dues à certaines protéines ou ARNs non traduits sans détailler davantage ces phénomènes. Des régulations sont présentes à chaque étape nécessaire à la production d'une protéine mature. Des composés régulateurs peuvent ainsi agir directement sur le brin d'ADN lors de la transcription, sur l'ARNpm durant sa maturation et sa traduction ou encore sur la protéine lors des modifications *post-traductionnelles*. Ces composés sont généralement formés de protéines et d'ARNs non traduits. De plus ces régulations peuvent avoir un effet tout autant positif que négatif sur l'expression d'un gène, on parle alors respectivement d'*activation* et d'*inhibition* de l'expression.

Régulations lors de la transcription Les premières régulations interviennent durant la transcription du brin d'ADN en ARNpm. Ces régulations proviennent de protéines, les *facteurs de transcription* (FT), qui viennent réguler la transcription en se fixant au niveau des promoteurs du gène. Trois types de promoteurs coexistent, un promoteur peut donc être central, proximal ou distal (voir Figure **5.2**). Le rôle du promoteur central est essentiel car c'est dans cette région que les FT se fixent afin de permettre à la fois l'initiation de la transcription mais aussi sa régulation. D'autres FT plus spécifiques viennent se fixer sur les promoteurs proximaux situés à une faible distance du promoteur central. Ces FT modulent également la régulation du gène associé. Le rôle des promoteurs distaux est quant à lui plus incertain, dû a leur éloignement du gène qu'ils régulent. Cependant en dépit de cette distance, le repliement de l'ADN permet de rapprocher certains de ces promoteurs qui peuvent alors interagir en présence des FT associés.

Il existe également des régulations dites *épigénétiques* dans le sens où celles-ci ne dépendent

pas d'une modification de la séquence d'ADN. L'une de ces régulations dépend de l'accessibilité de la séquence d'ADN à l'enzyme permettant de synthétiser l'ARNpm. Nous avons vu que l'ADN est pelotonné autour de protéines formant les *nucléosomes*, la structure de ces *nucléosomes* est caractérisée selon la *chromatine*, présente sous deux formes possibles. Lorsque la *chromatine* est condensée (*hétérochromatine*) l'enzyme ne peut s'accrocher au promoteur central, empêchant ainsi la transcription. A l'inverse lorsque la *chromatine* est dé-condensée (*euchromatine*) la transcription peut se dérouler. La chromatine est généralement sous sa forme compacte au sein d'une cellule empêchant la traduction d'un grand nombre de gènes. La modification de la chromatine et donc de l'expression des gènes s'effectue via des changements dans la composition des *histones* (complexes de protéines).

Régulations post-transcriptionelles Les ARNpm lors de leur maturation sont également soumis à certaines régulations venant de protéines mais aussi d'autres ARNs. Reprenons l'exemple de la phase d'épissage qui consiste à retirer les introns de l'ARNpm. Ce processus requiert la fixation d'un complexe, le *spliceosome* composé notamment de plusieurs protéines. Leur présence ou leur absence dans la cellule modifie alors les limites des régions à retirer, ce qui provoque l'apparition d'épissages alternatifs. Nous avons également mentionné le cas des ARNs non traduits tel les micro-ARNs qui inhibent la traduction d'un ARN mature cible. Le micro-ARN se fixe sur l'ARN cible grâce à sa séquence complémentaire à celle de sa cible. La traduction est d'autant plus inhibée que les séquences entre du micro-ARN et de sa cible sont similaires pouvant aller jusqu'au blocage complet de la traduction.

Régulations lors de la traduction De même que l'épissage, la phase de traduction nécessite un complexe, le *ribosome*, composé d'ARN ribosomiques. La réduction du niveau de ces ARNr limite le nombre de ribosomes entrainant une inhibition de la traduction. D'autres régulations sont issues de la fixation de protéines sur l'ARNm entre la coiffe et le codon indiquant le démarrage de la traduction, empêchant ainsi cette phase.

Régulations post-traductionelles Les modifications de la chaine d'acides aminés synthétisée lors de la phase de traduction dépendent à nouveau d'un ensemble de protéines. Ainsi certaines d'entre-elles régulent le niveau d'une protéine cible en modifiant sa structure changeant ainsi son rôle dans la cellule. Tandis que d'autres se combinent entre elles afin de former des éléments fonctionnels. Nous noterons que le terme régulation est utilisé ici de façon abusive pour caractériser ce qui relève dans certains cas davantage d'une interaction entre les protéines que d'une réelle régulation.

5.2.2 Les différents types de réseaux de régulation

Nous venons de voir qu'il existe des régulations à tous les niveaux du processus de synthèse des protéines impliquant à la fois des protéines, des ARNs et la séquence d'ADN. L'ensemble de ces régulations peut être représenté sous forme de graphe où chaque nœud représente une entité biologique distincte (gène, ARN, protéine) et les arcs entre ces nœuds symbolisent les régulations présentes. On parle alors de réseau de régulation biologique dont un exemple est présenté sur la Figure **5.3**. La prise

en compte de l'ensemble de ces régulations permet d'appréhender au mieux le fonctionnement réel d'un organisme. Cependant l'observation conjointe des 3 acteurs protéines/ARN/ADN est complexe et nécessite des ressources techniques et humaines importantes. Nous avons déjà évoqué cette limite lors de la définition du niveau d'expression des gènes. Le choix entre le taux d'ARNs ou de protéines synthétisés mène à la construction de deux types de réseaux : les réseaux de protéines et les réseaux transcriptionnels où chaque nœud représente respectivement une protéine spécifique ou un ARN distinct.

Les réseaux de protéines Les protéines représentent le dernier niveau de l'expression d'un gène, les interactions et régulations observées entre ces protéines informent donc de leur variation effective au sein de la cellule. Cependant ce niveau d'observation ne permet pas de représenter certaines régulations telles que celles issues d'ARN non-traduits. Alors même que ces ARNs peuvent être la cause de certaines variations du niveau de protéines, ceux-ci ne peuvent être représentés dans ce réseau.

Les réseaux transcriptionels La mesure des taux de transcrits permet au contraire de représenter ces régulations dues aux ARNs, que ceux-ci soient effectivement traduis ou non. En contre-partie, de la même manière que certaines régulations échappent aux réseaux de protéines, la mesure des niveaux de transcrits ne permet pas d'observer les régulations se déroulant pendant et après la traduction. Ces réseaux ne peuvent donc pas rendre compte des variations effectives des niveaux de protéines dans la cellule.

Régulations directes et indirectes Nous venons de mentionner que les réseaux transcriptionnels (de même que les réseaux de protéines) ne permettent pas de représenter toutes les régulations, analysons alors les conséquences de cette limite pour la situation présentée sur la Figure **5.3**. Nous supposerons que toutes les régulations présentes sont activatrices. La régulation (1) issue de la protéine A n'est observable qu'au travers des variations de quantité de la protéine B et non pas de l'ARNm. Ainsi lorsque le niveau de transcrits du gène A augmente, on observe une augmentation des quantités de protéines A et B, tandis que le niveau de transcrits du gène B n'est pas modifié. La régulation (1) est donc invisible au niveau transcriptionel.

De même si aucun facteur extérieur n'engendre d'augmentation du niveau de transcrits du gène B celui-ci stagne alors même qu'une augmentation de la quantité de protéines B, provoquée par la régulation (1), engendre une augmentation du taux de transcrits du gène C. La régulation (2) est donc également invisible au niveau transcriptionnel. Ces deux régulations ne peuvent donc pas être détectées de manière *directe*. On note cependant qu'à ce niveau, une régulation plus globale du gène A vers le gène C est détectable. On parle alors de régulation *indirecte* car l'augmentation du taux de transcrit du gène A induit une augmentation de celui du gène C. Dans ce cas l'information liée au gène B est perdue, malgré tout, la régulation indirecte résultante garde tout son sens, cette situation est équivalente à l'absence d'observations pour le gène B.

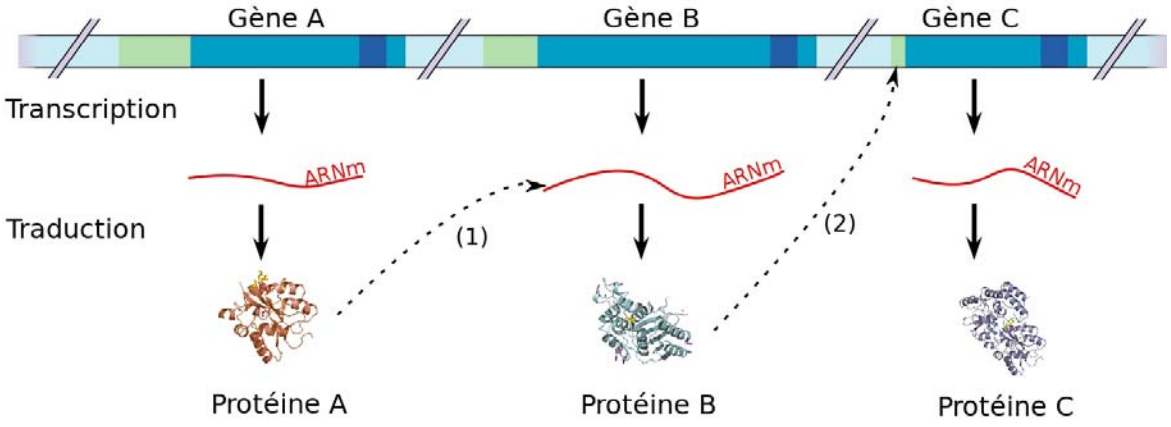


Figure 5.3 – Réseau de régulation biologique impliquant 3 gènes ainsi que les ARN et protéines synthétisées. Les flèches en traitillé représentent les phénomènes de régulation. Deux types de régulations sont représentées. La protéine A se fixe sur l'ARNm synthétisée par le gène B régulant ainsi sa traduction tandis que la protéine B se fixe au niveau du promoteur du gène C régulant sa transcription.

5.2.3 Les réseaux de régulation de gènes

"Quels sont les régulateurs de ce gène d'intérêt ?" ou "Quels gènes sont régulés par ce gène d'intérêt ?" sont deux questions récurrentes dans le domaine de la génétique. Un *réseau de régulation de gène* (noté RRG) est un concept générique dans lequel chaque nœud du réseau est assimilé à un gène et chaque arc représente une régulation d'un gène sur l'expression d'un autre gène. Cependant du fait que les régulations entre gènes n'interviennent qu'au travers des molécules synthétisées, à savoir les ARNs et les protéines, il est nécessaire de spécifier laquelle de ces deux molécules permet de capter au mieux l'expression globale d'un gène. Cette question revient à se demander lequel des réseaux de protéines ou transcriptionnel représente au plus près un RRG. L'utilisation des taux de transcrits afin de caractériser cette expression semble la plus pertinente du fait que la bijection 1 gène $\longleftrightarrow$ 1 ARN n'est remis en cause qu'en cas d'épissage alternatif. Alors que le rapprochement entre les réseaux de protéines et les RRG est plus délicate du fait qu'un même gène synthétise bien souvent plusieurs protéines distinctes. Bien qu'aucun de ces deux réseaux ne représente exactement un RRG au sens le plus strict, c'est-à-dire que chaque nœud modélise l'intégralité des produits d'un gène, les réseaux transcriptionnels apparaissent comme la meilleure des deux alternatives. Par la suite nous utiliserons donc le terme RRG dans le sens des réseaux transcriptionnels.

Reconstruction des RRG Trouver le RRG d'un organisme est une problématique de recherche très active ces dernières années dont les applications possibles sont nombreuses, notamment au

niveau thérapeutique. Les premières méthodes automatiques développées dans le but d'apprendre la structure de ces RGG utilisent les corrélations entre les niveaux d'expression afin d'en déduire des régulations entre gènes. Plus récemment des approches dites *intégratives* utilisent en plus de ces données d'expression, des informations sur la séquence génomique afin d'améliorer la reconstruction du réseau. Par exemple, la détection de sites d'accroches à certaines protéines permet de fournir un *a priori* sur les régulations existantes. Nous présentons dans la section suivante une autre source d'information basée cette fois sur les mutations observées de la séquence d'ADN.

5.3 Polymorphismes et variations d'expression

Des mutations spontanées de certains nucléotides sur la séquence d'ADN peuvent être la source de variations dans l'expression des gènes. Chaque mutation correspond au changement d'une base dans la séquence d'ADN, on parle alors de SNP (Single Nucleotide Polymorphism). La présence de SNPs sur le génome engendre des situations de polymorphismes génétiques où plusieurs versions d'un même gène, les *allèles*, coexistent au sein d'une population d'individus. Chaque *allèle* d'un gène diffère d'un seul ou de plusieurs nucléotides dans la séquence codante. Ces polymorphismes impliquent des variations dans l'expression des gènes et peuvent être à l'origine de maladies génétiques ou de modifications d'un caractère physique pour un individu (couleur de la fourrure chez l'animal par exemple). Certains cas de polymorphisme peuvent être complexes et dépendent d'une configuration d'*allèles* spécifique portant sur plusieurs gènes.

Régulations *cis* et *trans* La position du SNP sur la séquence conditionne l'effet de celui-ci sur l'expression des gènes, nous pouvons ainsi distinguer deux positions ayant des effets distincts. La première situation correspond à la présence d'une mutation dans une des régions promotrices du gène, le SNP perturbe alors la fixation des facteurs de transcription au niveau de cette région et donc leur capacité de régulation. Dans ce cas le SNP a un impact direct sur le niveau d'expression du gène correspondant, on dit alors que le polymorphisme a un effet en *cis* sur ce gène.

Dans la seconde situation la mutation a lieu dans la région codante du gène G_i, la séquence d'ARN traduite puis la structure de la protéine potentiellement synthétisée sont alors modifiées. Dans ce cas le niveau d'expression de G_i n'est pas changé, c'est-à-dire que les quantités d'ARNs et de protéines synthétisés restent les mêmes. Les changements de composition de l'ARN et de la protéine altèrent de fait leur efficacité en tant qu'acteurs de la vie cellulaire et notamment en tant que régulateurs d'autres gènes. Supposons que G_i régule un gène G_j, l'expression de G_j sera alors modifiée suite à cette mutation. Le SNP présent dans la région codante de G_i a donc un impact sur le niveau d'expression des gènes, tels que G_j, régulés par G_i, on dit ici que le polymorphisme a un effet en *trans* sur G_j. On entrevoit ici l'intérêt d'une analyse conjointe des variations d'expressions dues aux polymorphismes et de leur position sur la séquence afin de mettre à jour certaines régulations entre gènes. Un troisième type de régulation est parfois défini lorsque un SNP situé dans la région promotrice d'un gène G_i (relation en *cis*) régule à son tour un gène G_j. Cette relation indirecte (via G_i) du polymorphisme vers G_j est qualifiée de régulation *cis-trans*.

Afin de distinguer expérimentalement l'effet en *cis* d'un SNP d'un effet en *trans*, il est possible d'utiliser une recherche d'eQTL (expression Quantitative Trait Locci). Cette analyse est une extension de la recherche de QTL qui vise à rechercher pour une variable quantitative les régions du génome dont les variations, observées au travers des SNP, permet d'expliquer les variations de la variable d'intérêt. Dans le cas des eQTLs la variable quantitative correspond à l'expression d'un gène. Les régions détectées sont de tailles variables en fonction de la corrélation plus ou moins forte entre la variable d'expression et les variations de la séquence. Cette corrélation est quantifiée à l'aide d'un test de rapport de vraisemblance, le score LOD. Ainsi un score LOD élevé indique une corrélation forte ce qui mène à la détection de courtes régions très piquées tandis qu'un score LOD faible induit des régions de confiance plus larges et moins piquées. Lorsque le pic de l'une des régions détectées pour un gène coïncide avec la position de ce gène sur le génome, alors le SNP proche de ce pic sera défini comme ayant un effet en *cis*, à l'inverse des régions éloignées du gène permettront de classer les SNPs présents dans ces régions comme ayant un effet potentiel en *trans*.

Cependant tous les SNPs n'engendrent pas ce type d'effet, ainsi la présence d'un SNP dans des régions non codantes ou n'ayant aucun rôle régulateur n'a *a priori* aucun impact sur l'expression des gènes. De même certaines mutations au niveau des *exons* ou des promoteurs peuvent être silencieuses dû au caractère dégénéré des séquences. La dégénérescence des régions promotrices s'explique du fait que certains facteurs de transcription ne requièrent pas une séquence parfaite du promoteur afin de s'y accrocher, des mutations peuvent alors avoir lieu sans perturber leur fixation. Au niveau des *exons* c'est le caractère redondant de certains codons qui amène à la dégénérescence, ainsi à plusieurs codons différents (successions de 3 nucléotides) correspond un même acide aminé, une mutation peut ainsi modifier un codon sans pour autant changer l'acide aminé généré et donc la protéine synthétisée.

Hormis ces situations de polymorphisme silencieux, l'information des SNPs obtenue par génotypage associée à celle de leur position par rapport aux gènes représente une source supplémentaire d'information aidant à représenter plus finement les RRG. On notera cependant que si la position des gènes est souvent définie avec précision sur le génome, les limites des régions promotrices associées le sont plus rarement. La combinaison de données d'expression et d'informations liées à la séquence du génome au travers de SNP a donné lieu à l'émergence de nouvelles méthodes de reconstruction des RRG tirant partie de cette source d'information combiné dite de *génétique-génomique*.

5.4 Conclusion

Ce chapitre introduisant les concepts biologiques liés à l'expression des gènes a permis d'entrevoir la complexité des phénomènes de régulation qui entrent en jeux au sein de la cellule. Cette complexité provient à la fois de la variété des régulations agissant aux différents stades de l'expression d'un gène et des différents acteurs de ces régulations. Reconstruire le réseau d'interactions dans son intégralité est une tâche encore trop complexe, à la fois par le nombre de variables mis en jeu mais aussi par le protocole expérimental nécessaire afin d'obtenir les données biologiques correspondantes. A défaut de

reconstruire un tel réseau les méthodes actuelles de reconstruction se focalisent sur un unique niveau de représentation des régulations. Nous avons présenté ici deux niveaux de représentation possibles : les réseaux de protéines et les réseaux transcriptionnels. Ce dernier représente l'alternative la plus proche du concept même de réseau de régulation de gène permettant de répondre à la question "Quel gène régule tel autre gène ?". Cependant toutes les régulations existantes ne peuvent être représentées par le réseau transcriptionnel, certaines d'entres elles n'étant pas transposables au seul niveau des ARNs. Cette restriction génère alors des régulations indirectes dans le réseau, qui, sans être erronées, font perdre en qualité de description du phénomène biologique réel.

De nouvelles approches dites *intégratives* émergent progressivement afin d'utiliser conjointement diverses sources d'information en plus des données d'expression. Le but étant de reconstruire au mieux les RRG. La difficulté supplémentaire résulte alors de l'intégration de ces données hétérogènes au sein d'un même modèle. Dans le chapitre 7 nous présenterons une modélisation intégrative basée sur les réseaux bayésiens afin d'exploiter des données combinées d'expression de gènes et de polymorphisme, dites de *génétique-génomique*.

Chapitre 6

Apprentissage d'un réseau de régulation de gènes

Sommaire

6.1 Caractéristiques des données 116
6.1.1 Observations statiques versus temporelles 116
6.1.2 Données discrètes versus continues 117
6.1.3 Variabilité des données d'expression 118
6.2 Approches existantes à partir de données d'expression 120
6.2.1 Méthodes non-paramétriques 121
6.2.1.1 Dépendances statistiques 121
6.2.1.2 Arbres de régression 124
6.2.2 Modèles graphiques gaussiens 125
6.2.3 Réseaux bayésiens 130
6.3 Approches intégratives 131
6.4 Conclusion 134

Nous verrons dans ce chapitre les différentes méthodes développées afin d'apprendre les réseaux de régulation de gènes. Certaines d'entre-elles utilisent des données spécifiques. Nous définissons donc dans un premier temps les éléments qui caractérisent les données d'expression et les moyens permettant d'obtenir une variation de celles-ci. Nous nous concentrons par la suite sur les approches qui utilisent des données d'expressions statiques, nous décrivons à la fois les différentes modélisations possibles du problème de reconstruction des RRG et les algorithmes proposés pour chacune d'elles. Nous terminons enfin par une présentation des développements récents des méthodes dites *intégratives* qui utilisent diverses informations biologiques en plus des données d'expression afin d'améliorer la qualité de la reconstruction. Nous nous focalisons principalement pour ce panorama sur les méthodes qui utilisent des données de *génétique-génomique*.

6.1 Caractéristiques des données

La méthode d'apprentissage d'un RRG dépend du type de données employées. Nous avons vu dans le chapitre précédent que le niveau d'ARN transcrits dans la cellule était la mesure la plus couramment employée afin de caractériser le niveau d'expression des gènes. Cependant le protocole expérimental mis en place afin de collecter ces niveaux de transcrits amène à la production de différents types de données. La première distinction s'effectue suivant la prise en compte du caractère temporel des observations.

6.1.1 Observations statiques versus temporelles

La mesure des niveaux de transcrits que celle-ci soit effectuée grâce aux puces à ADN ou par séquençage, correspond à l'état d'un ensemble de cellules à un instant donnée, comme une photographie à un instant t des quantités d'ARN présentes dans l'échantillon analysé. Ces mesures sont alors considérées comme représentant l'état stable des différents gènes au sein de la cellule, cette vision reste néanmoins réductrice du fait de l'évolution permanente de l'activité des gènes. Ainsi différentes mesures effectuées pour un même individu et dans les même conditions peuvent varier de manière naturelle. L'un des moyens pour limiter cette variabilité consiste à utiliser des échantillons composés de plusieurs cellules, ce qui permet alors de définir une activité moyenne des gènes pour un type de cellule donné. De plus, du fait que l'ensemble des gènes exprimés diffère en fonction du type de la cellule, les cellules analysées proviennent des mêmes tissus.

Lorsque les cellules sont observées dans un système stable, sans perturbation, que l'on peut supposer à l'équilibre, on observe un seul point temporel et l'on parlera de données *statiques*.

Cependant il est également possible d'effectuer différentes mesures espacées dans le temps afin d'obtenir des données dites *temporelles*. Dans ce cas il convient de fixer un intervalle de temps entre chaque mesure, ce paramètre est essentiel et dépend du type de variation que l'on souhaite observer.

L'utilisation de données temporelles afin d'analyser un système dynamique tel que la cellule semble naturelle, d'autant plus que ces observations permettent d'orienter des relations causales entre les gènes. Cependant la production de données temporelles est complexe, le caractère temporel des mesures nécessite une grande précision de l'exécution du protocole expérimental ce qui restreint généralement le nombre de mesures successives à quelques dizaines pour un individu. A contrario les populations utilisées afin de produire des données statiques peuvent atteindre plusieurs centaines d'individus et donc autant d'observations. Cette différence au niveau de la complexité expérimentale explique la généralisation des méthodes utilisant des données statiques au dépend des données temporelles. Pour cette même raison nous nous sommes placés dans cette thèse dans le cadre de l'analyse de données statiques pour lesquelles nous détaillerons les méthodes existantes en section 6.2. Afin de donner quelques idées des méthodes dédiées aux données temporelles, nous présentons brièvement dans la suite de cette section, deux modélisations fréquemment employées.

Méthodes de reconstruction à partir de données temporelles La première de ces modélisations est basée sur le formalisme des réseaux bayésiens dynamiques présentés brièvement dans le chapitre 1. Cette modélisation permet de lever la contrainte d'acyclicité présente dans le cas de données statiques. De nombreux travaux utilisant cette modélisation ont été développés afin d'analyser des données d'expression de gènes où chaque couche du graphe représente les observations à un instant donné et chaque arc entre ces couches, une régulation.

Une seconde approche consiste à définir un système d'équations différentielles afin de décrire la dynamique des niveaux de transcrits au sein de la cellule. L'activité de chaque gène G_i est alors modélisée par une équation différentielle de type

$$\frac{dG_i}{dt} = f(G_j)$$

où G_j représente les niveaux de transcrits des gènes régulant G_i au travers d'une fonction f.

La nature de la fonction f permet de représenter des interactions complexes entre les niveaux d'expression des gènes régulateurs, comme le cas des complexes protéiques qui agissent de concert afin de former des facteurs de transcription. Cette représentation permet de représenter au plus près le phénomène de régulation, il est cependant nécessaire de connaître à l'avance le type de régulation attendue afin de définir une formulation mathématique adéquate. De plus la complexité de cette modélisation à un coût en ce qui concerne la phase d'apprentissage de la structure, en effet apprendre les paramètres d'un modèle non linéaire est complexe. De manière générale les modèles non linéaires sont utilisés dans un but de simulation de l'activité d'un réseau défini à l'avance. Dans le cas de l'apprentissage de la structure du réseau, des fonctions linéaires leurs sont préférées du fait d'une estimation simplifiée des paramètres.

Nous n'aborderons pas davantage ces approches, plusieurs états de l'art récents de ces techniques ayant été effectués par Sima et al. [2009], Pal et al. [2012]. Nous noterons pour conclure que certaines modélisations se satisfont aussi bien de données temporelles que statiques comme les méthodes basées sur les arbres de décision ou les modèles graphiques gaussiens. Approches que nous décrirons un peu plus loin dans le cas des données statiques.

6.1.2 Données discrètes versus continues

La deuxième caractéristique discriminante des données d'expression est la nature continue ou discrète des observations. Cette distinction dépend du protocole expérimental ayant servi à les générer. L'utilisation de puces à ADN permet d'obtenir des observations continues correspondant aux mesures d'intensités lumineuses (ou radioactives) des spots. Les technologies de séquençage génèrent quant à elles des données discrètes correspondant aux comptages des séquences d'ARN présentes dans l'échantillon cellulaire.

Cependant les données continues issues des puces à ADN peuvent être discrétisées, afin de réduire le bruit expérimental inhérent à des mesures d'intensités. Les méthodes de discrétisation telles que celles présentées dans la section 2.2 sont alors utilisées afin d'obtenir un nombre limité de classes (≤ 5). Les données générées par le séquençage bien que discrètes subissent également

une transformation similaire afin de réduire le nombre de classes. Cette restriction est nécessaire afin de limiter la complexité algorithmique de certaines approches, notamment dans le cadre des RB où le nombre de configurations possibles des parents pour une variables augmente de manière exponentielle en fonction du nombre de classes des variables impliquées. De plus comme nous l'avons déjà mentionné, l'interprétation biologique d'un nombre limité d'états (typiquement 3) est plus aisée. La question de la perte d'information due à cette simplification des données est alors posée de même que l'interprétation à donner à la discrétisation effectuée. C'est dans ces deux optiques que nous avons à la fois proposé une discrétisation visuelle adaptée aux données d'expression ainsi qu'une discrétisation permettant de limiter cette perte d'information.

Le choix durant la thèse d'utiliser le formalisme des réseaux bayésiens discrets nous amène naturellement à ne considérer que les données discrètes et nécessite en toute logique une phase de discrétisation des niveaux d'expressions. Cependant certaines approches ne requièrent pas cette étape et utilisent les données d'expression sous leur forme continue. Nous introduirons donc dans la section 6.2 les différentes approches de reconstruction de RRG existantes indépendamment de la nature discrète ou continue des données utilisées.

6.1.3 Variabilité des données d'expression

Afin de reconstruire un RRG de bonne qualité il est primordial d'effectuer un nombre conséquent d'observations. De plus les variations entre ces observations doivent être significatives et ne doivent pas être seulement dues au bruit expérimental. Dans le cas des données statiques, chaque observation correspond à la mesure des niveaux de transcrits d'un ensemble de gènes pour un individu. Certaines des variations observées entre individus sont provoquées par les mutations génétiques se produisant au sein d'une même espèce, comme nous l'avons expliqué précédemment. L'utilisation de cette variabilité génétique des individus afin d'obtenir une variation des niveaux de transcrits est donc une approche parfaitement adaptée au cadre des données de *génétique-génomique* où ces deux sources de variabilités sont justement observées. Dans ce cas l'utilisation d'une population contrôlée est toute fois nécessaire, comme nous l'expliciterons dans le paragraphe suivant.

Une autre technique permettant d'observer une variation d'expression consiste à perturber expérimentalement les individus. Ces perturbations peuvent être appliquées de manière externe ou interne. Les perturbations externes tel qu'une exposition à des rayonnements, à des composées chimiques ou tout autre perturbation de son environnement provoque un modification de l'activité cellulaire et donc de l'expression des gènes. De manière générale ces réactions de défense ou d'adaptation de la cellule induisent une variation dans l'expression de nombreux gènes à la différence des perturbations internes. Ces dernières sont ciblées sur un gène spécifique afin de moduler son activité à l'aide d'ARN spécifiques ou de supprimer intégralement ce gène des chromosomes en remplaçant la séquence d'ADN correspondante par une séquence non-codante, on parle alors de perturbations *knock-out*. Ces interventions ciblées modifient l'expression du gène cible et perturbent ainsi l'ensemble de la chaîne de régulations dans laquelle ce gène est impliqué. Ce type de perturbation est donc idéal afin d'étudier un sous ensemble de gènes d'intérêt et de reconstruire spécifiquement un sous-réseau de régulation entre ces gènes.

Population contrôlée en génétique-génomique Afin d'expliquer la spécificité des données employées en *génétique-génomique*, retournons au niveau de l'ADN et de son organisation en chromosome. Le nombre de chromosomes varie entre chaque espèce mais également le nombre d'exemplaires de chacun de ces chromosomes dans la cellule. Pour la majorité des organismes chaque chromosome est présent, en un seul ou deux exemplaires regroupés dans ce cas sous forme de paire, on parle alors respectivement de cellules *haploides* ou *diploides*. Plus rarement certaines cellules dites *polyploides* possèdent un nombre d'exemplaires de chaque chromosome plus élevé. Nous nous intéressons plus particulièrement au cas des organismes dont les cellules sont *diploides* comme chez la plupart des animaux ainsi que pour une partie des végétaux.

Hormis le cas des chromosomes sexuels, les deux chromosomes d'une même paire, dits homologues, contiennent le même ensemble de gènes. La séquence d'ADN du gène peut cependant varier entre les deux chromosomes homologues, les mutations touchant indépendamment un seul des deux chromosomes. On obtient donc deux versions d'un même gène, appelées les *allèles*, plusieurs allèles d'un même gène coexistent fréquemment au sein d'un organisme sans compromettre sa viabilité. Dans ce cas le gène est dit *hétérozygote*, a contrario lorsque la séquence du gène est identique entre les deux chromosomes homologues, celui-ci est dit *homozygote*. Une hiérarchie entre allèles d'un même gène peut exister au sein de l'activité cellulaire ce qui conduit à des situations où l'expression d'un allèle présent sur l'un des deux chromosomes domine le second. Cette situation de dominance complexifie la relation entre les polymorphismes observés et l'expression des gènes car l'effet d'un polymorphisme observé sur un chromosome se trouve être conditionné par l'allèle présent sur le second chromosome. Pour cette raison dans le cadre des données de *génétique-génomique*, il est plus simple d'utiliser des observations issues d'individus pour lesquels l'ensemble des gènes des cellules étudiées soit homozygotes.

Des populations d'individus homozygotes sont alors obtenus par croisements *contrôlés* de type *RIL* (*Recombinent Inbred Lines*) ou *lignée recombinante consanguine*. Ce type de lignée, illustrée par la Figure **6.1**, permet à partir de deux individus homozygotes présentant une forte variation au niveau de leurs allèles, d'obtenir un ensemble d'individus homozygotes eux aussi ayant un patrimoine génétique constitué d'une mosaïque des allèles des deux parents fondateurs. Les différences génétiques entre les deux fondateurs s'expliquent par les mutations successives apparaissant chez une espèce au cours de l'évolution, donnant naissance à différentes familles au sein de cette espèce présentant des profils génétiques variés. On obtient à partir d'un premier croisement entre les deux parents fondateurs P1 et P2, un couple d'individus hétérozygotes représentant la génération F1. Ces deux individus servent à leur tour à générer une population F2 constituée d'un nombre paire d'individus, toujours hétérozygotes. Nous rappellerons qu'à chaque croisement l'individu généré hérite d'un chromosome de chacun de ses parents mais que des situations dites de *crossing-over* mènent à ce que le chromosome hérité d'un des parents soit une combinaison des deux chromosomes de celui-ci. Les individus de la famille F2 sont ensuite regroupés par paire afin que chacune d'elles donne naissance à une lignée spécifique. Cette lignée se développe par croisements successifs des deux individus obtenus à chaque génération. Dans le cas des plantes il est possible de simplifier ce schéma en croisant à chaque génération un individu avec lui même sans avoir recours au croisement entre frères, on parle alors

d'*autofécondation*. Ces croisements se répètent jusqu'à obtenir des individus homozygotes. Cette convergence est assurée par le fait que le caractère homozygote d'une séquence d'ADN correspond à un état puits dans la théorie des graphes d'où il est impossible de revenir à un état d'hétérozygotie. Le nombre de générations nécessaire à la convergence est relativement faible, dès la 7ème génération pour les schémas d'autofécondation, les individus possèdent une proportion de séquences génétiques hétérozygotes négligeable comparée à celle de séquences homozygotes.

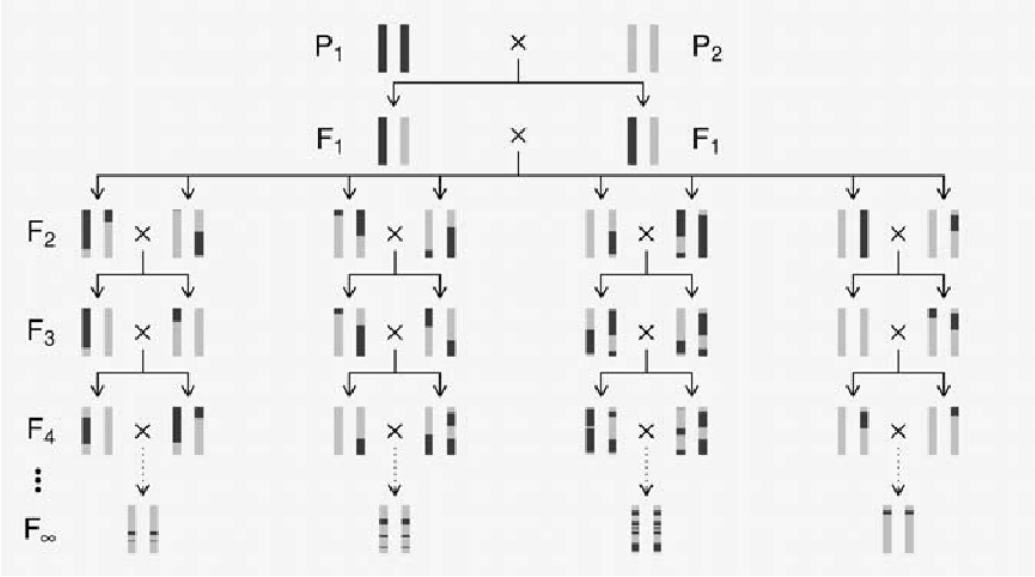


Figure 6.1 – Lignée recombinante consanguine par croisement entre frères (RIL).

6.2 Approches existantes à partir de données d'expression

Nous présentons ici les différentes méthodes proposées afin de reconstruire un RRG dans le cas de données statiques.

Jusqu'à aujourd'hui, la majorité des méthodes développées afin de reconstruire les RRG utilisent principalement des données d'expression. Certaines méthodes reposent sur des concepts simples tandis que d'autres proposent une modélisation statistique du problème. Ces dernières posent alors certaines hypothèses comme l'existence d'une distribution sous-jacente aux données. Ce qui ne signifie pas pour autant que ces méthodes ne sont applicables qu'aux données suivant une telle distribution, ce qui empêcherait alors leur application aux données réelles qui suivent rarement une distribution précise. Nous présenterons tout d'abord des méthodes dites *non-paramétriques* qui ne nécessitent justement pas ce type d'hypothèse sur les données.

6.2.1 Méthodes non-paramétriques

6.2.1.1 Dépendances statistiques

Apprendre la structure d'un réseau de régulation revient à répondre à la question "Quels sont les gènes G_j qui régulent l'expression de G_i ?". Or le fait qu'un gène G_j régule G_i induit que leurs niveaux de transcrits évoluent de manière coordonnée, autrement dit, qu'il existe une dépendance entre les niveaux d'expression des gènes liés par des régulations. Rechercher les gènes dont l'expression est dépendante de celle de G_i permet donc de trouver les gènes le régulant. Parmi les mesures de dépendances statistiques fréquemment employées nous pouvons citer l'information mutuelle que nous avons déjà présenté dans la partie 1 mais aussi le calcul des coefficients de corrélations dans le cas de dépendances linéaires entre deux variables.

Citons par exemple la corrélation de Pearson dont les coefficients sont définis par

$$r(G_i, G_j) = \frac{cov(G_i, G_j)}{\sqrt{var(G_i)\ var(G_j)}}$$

où $r(G_i, G_j)$ représente le coefficient de corrélation entre les gènes G_i et G_j correspondant à une normalisation de la covariance $cov(G_i, G_j)$ entre les deux gènes par le produit de leur variance. Cette corrélation signée prend ses valeurs dans l'intervalle [-1 ;1] permettant de distinguer les régulations activatrices et inhibitrices. Lorsqu'il existe une forte dépendance entre les deux gènes, le coefficient associé se rapproche en valeur absolue de 1 tandis que leur indépendance se traduit par un coefficient nul. Il faut cependant noter que $r_{G_i,G_j} = 0$ n'implique généralement pas que G_i et G_j soient indépendants, cette propriété n'étant vraie que pour deux variables suivant une loi normale et liées entre elles par une fonction linéaire.

Afin de reconstruire le RRG il suffit donc de mesurer pour chaque gène le degré d'indépendance avec chacun des autres gènes grâce à une mesure adéquate et de définir le seuil à partir duquel la dépendance est jugée suffisante pour supposer l'existence d'une régulation. Afin de fixer ce seuil des techniques de rééchantillonnage telles que le *bootstrapping* peuvent être utilisées pour maintenir un niveau de signification sur les dépendances retenues. Malgré une estimation adéquate de ce seuil le nombre de régulations apprises à tort reste élevé. En effet, les mesures de dépendances simples ne permettent pas de distinguer les relations directes ($G_j \rightarrow G_i$) des relations indirectes via un 3$^{\text{ème}}$ gène G_k ($G_j \rightarrow G_k \rightarrow G_i$) ni des situations de co-régulation ($G_i \leftarrow G_k \rightarrow G_j$). Dans ces deux dernières situations G_i et G_j sont détectées comme dépendants alors qu'aucune régulation directe n'existe réellement. De manière générale le terme de réseau de *co-expression* est préféré dans ce cas au terme de RRG du fait que les mesures de dépendance simple permettent uniquement de définir des groupes de gènes dont les variations d'expression sont corrélées.

Dépendances partielles Afin de supprimer les relations indirectes l'une des possibilités, déjà mentionnée dans la partie 1, consiste à utiliser des mesures de dépendances partielles, comme l'information mutuelle conditionnelle. De la même manière il est possible de définir des mesures de corrélation partielle.

La corrélation partielle de Pearson d'ordre 1 s'écrit alors

$$r(G_i, G_j|G_k) = \frac{r(G_i, G_j) - r(G_i, G_k)r(G_j, G_k)}{\sqrt{1 - r(G_i, G_k)^2}\ \sqrt{1 - r(G_j, G_k)^2}}$$

où $r(G_i, G_j)$ correspond à la corrélation dite *brute* définie précédemment et G_k représente la variable de conditionnement dans le cas gaussien ou plus généralement appelée variable de *contrôle*.

Une unique variable de contrôle ne permet pas systématiquement de détecter l'indépendance entre deux gènes, dans ce cas on étend la notion de corrélation partielle à l'ordre k où l'ensemble de conditionnement est constitué de k variables de contrôle.

Ainsi dans le cas d'une structure en chaîne ($G_j \rightarrow G_k \rightarrow G_i$) ou divergente ($G_i \leftarrow G_k \rightarrow G_j$), l'indépendance d'ordre 1 entre G_i et G_j sachant à G_k peut être détectée ($r(G_i, G_j|G_k) \approx 0$ dans le cas gaussien). Afin de supprimer les relations indirectes entre deux gènes, nous devons calculer toutes les dépendances partielles pour cette paire sachant les ensembles de conditionnement possibles de taille k fixée. Si l'une de ces dépendances partielles est proche de 0 alors les deux gènes sont indépendants sachant l'ensemble de contrôle correspondant. La mesure de dépendance partielle d'ordre 1 entre deux gènes G_i et Gj s'écrit alors respectivement pour le coefficient de corrélation de Pearson et l'information mutuelle

$$Dep_r(G_i, G_j) = min_{k\neq i,j}(|r(G_i, G_j|G_k)|) \qquad et \qquad Dep_{IM}(G_i, G_j) = min_{k\neq i,j}(IM(G_i, G_j|G_k))$$

Ces expressions se généralisent pour un ordre k quelconque ($1 \leq k \leq p-2$).

Ordre des dépendances partielles Les mesures de dépendance partielle d'ordre k permettent donc de détecter les situations où deux gènes ont une frontière d'indépendance d'au plus k autres gènes dans le vrai RRG. Afin de couvrir toutes les relations indirectes possibles, il est donc nécessaire de considérer les ensembles de conditionnement pour chaque valeur possible de k. Ceci peut être fait de manière progressive en recherchant les indépendances d'ordre successif i à partir de $i = 0$ jusqu'à $i = p - 2$. A la fin de ce processus toute paire de gènes pour laquelle il existe au moins une mesure de dépendance partielle proche de 0 est jugée indépendante. Cependant, le nombre d'ensembles de conditionnement augmente de façon exponentielle avec k (C_k^{p-2}), l'énumération complète n'est donc possible que pour de faibles valeurs de k.

Une alternative consiste alors à considérer directement l'unique ensemble de conditionnement composé des $p - 2$ autres gènes qui permet donc théoriquement d'éliminer les relations indirectes et divergentes. Dans ce cas c'est le nombre d'observations qui devient facteur limitant. Nous savons que l'analyse des données d'expression constitue une situation dite de grande dimension, c'est-à-dire que le nombre de gènes est très en deçà du nombre d'observations, l'estimation des coefficients de corrélations ou de l'information conditionnelle n'est alors plus assez fiable. De plus l'utilisation directe des $p-2$ autres gènes comme variables de contrôle sans considérer les ensembles de conditionnement plus petits induit de fausses dépendances dans le réseau appris lors de l'existence de *v-structures*. En effet si G_k est régulé par deux gènes G_i et G_j, alors l'indépendance entre G_i et G_j n'est plus

maintenue dès lors que G_k appartient à l'ensemble de conditionnement, ce qui est forcément le cas lorsque $k = p - 2$.

Malgré ces faiblesses, les coefficients de corrélation partielle sont utilisés dans le cadre des modèles graphiques gaussiens. Nous détaillerons les méthodes qui en découle un peu plus loin dans ce chapitre.

Approches existantes Plusieurs approches ont été proposées afin de reconstruire des RRG en utilisant des ensembles de conditionnement limité. de la Fuente et al. [2004] proposent ainsi d'utiliser les corrélations partielles de Pearson conditionnellement à deux variables de contrôle au maximum. L'algorithme *ParCorA* suppose initialement que tous les gènes sont reliés entre eux puis calcule pour chaque paire l'ensemble des coefficients de corrélation partielle en augmentant progressivement la taille de l'ensemble de conditionnement. Dès qu'un de ces coefficients est inférieur à un seuil fixé, la relation correspondante est supprimée. L'algorithme repose sur le postulat que chaque gène ne subit d'influence notable que de deux cascades de régulations distinctes au maximum, et qu'inclure un gène de chacun de ces chemins dans l'ensemble de conditionnement, permet de retrouver la majorité des indépendances du vrai RRG. Ce postulat est cohérent avec le caractère peu dense des RRG. La corrélation de Spearman peut également être utilisée dans ce même but, cette mesure permet de se libérer de la contrainte de linéarité des relations en considérant le rang des observations plutôt que leur valeur exacte.

D'autres approches utilisent cette fois l'information mutuelle (IM) sans avoir nécessairement recourt à une mesure conditionnelle. Ainsi le logiciel *ARACNE* proposé par Margolin et al. [2006] débute par le calcul de l'ensemble des IMs par paire de gènes et retient uniquement les relations dont l'IM est supérieur à un seuil défini par permutation des données. A partir de cet ébauche de réseau l'algorithme analyse successivement tout les triplets de gènes formant une clique. Le principe de DPI (Data Porcessing Inequality) est alors testé pour chacun de ces triplets et vise à supprimer la relation dont l'IM est la plus faible parmi les trois relations composant la clique. Du fait que ce processus repose uniquement sur des comparaisons d'IM, celui-ci est peu sensible à la précision du calcul des IMs dès lors que le biais est commun à l'ensemble des gènes. Une condition supplémentaire permet de conserver les trois relations d'une clique si aucune des IMs est significativement plus faible. De même Faith et al. [2007] dans l'algorithme *CLR*, ne recourent pas aux mesures conditionnelles mais proposent de corriger l'IM d'une paire (G_i,G_j) en fonction de l'IM de chacun de ces deux gènes avec tous les autres gènes du réseau. Un nouveau score est ainsi proposé, basé sur la distance euclidienne de l'IM entre G_i et G_j avec les distributions de l'IM de (G_i,G_k) et de (G_j,G_k) pour chaque gène G_k du réseau. Ce nouveau score agit de la même manière qu'un test de significativité qui permet de sélectionner une relation qui se démarque du bruit de fond local aux deux gènes reliés. Le réseau final est obtenu grâce à un seuil sur ces IMs corrigées.

Ces méthodes basées sur la recherche des indépendances permettent de déterminer rapidement des solutions de bonne qualité, avec un nombre de paramètres réduit qui se résume bien souvent au seul seuil de significativité. Cependant du fait de la symétrie des mesures employées, les réseaux appris sont non-orientés. Aucune relation de régulation au sens strict ne peut donc en être extraite. Une seconde approche, toujours non-paramétrique, permet cette fois de déterminer un RRG complètement orienté

à l'aide d'arbres de régression.

6.2.1.2 Arbres de régression

Les arbres de régression tels que définis par Breiman et al. [1984] sont des outils de classification simple et efficace. Construire un arbre de régression pour une variable cible X_i équivaut au problème de sélection de variables explicatives pour X_i, c'est-à-dire des variables qui permettent de prédire l'état de X_i. Dans le cadre des RRG, les variables explicatives représentent typiquement les gènes régulant le gène cible G_i.

La construction de l'arbre de régression pour un gène cible G_i s'effectue par niveau dont le premier est constitué uniquement de la racine de l'arbre qui regroupe l'ensemble des observations de G_i. A partir de cette racine, deux nœuds fils sont créés par séparation des observations de G_i en une partition de deux sous-ensembles attribués à chacun des deux fils. Chaque nœud est donc défini par un sous-ensemble des observations de son unique parent. L'ensemble des nœuds formé à partir du niveau courant constitue le niveau suivant à partir duquel le processus se répète. A chaque nœud les observations de G_i sont séparées selon un gène discriminant G_j ($j \neq i$). Un seuil sur les observations de ce gène discriminant G_j est recherché de manière dynamique afin de créer un test binaire ($G_j > \Delta \mid G_j \leq \Delta$) qui coupe les observations de G_j (et donc de G_i) en deux sous-ensembles. Le choix du gène discriminant ainsi que celui du seuil associé est guidé par la minimisation de la variance des observations de G_i dans ces deux sous-ensembles. Il est alors possible de quantifier pour G_i l'importance d'un gène G_j en effectuant la somme des réductions de variance que G_j a permis lorsque celui-ci a servi de variable discriminante. Ainsi les gènes discriminants qui séparent les observations de G_i très tôt dans la construction de l'arbre, c'est-à-dire ceux qui réduisent fortement la variance des observations, représentent les gènes régulateurs les plus probables pour G_i.

Ce type d'arbre peut être construit pour chaque gène de manière indépendante afin d'obtenir un classement local des régulateurs potentiels. De plus la normalisation des observations permet d'obtenir un classement global de toutes les régulations possibles entre gènes. Cette méthode ne permet donc pas de construire à proprement parler un RRG mais de fournir un classement des régulations, afin d'obtenir un RRG il est donc nécessaire de définir un seuil de significativité.

Il est possible d'augmenter la robustesse de ce type d'approche grâce à la construction pour chaque gène cible d'un ensemble d'arbre en incluant un caractère stochastique. Ces méthodes permettent ainsi d'explorer différents arbres en perturbant l'ensemble d'observations ou en restreignant le choix des variables discriminantes possibles.

Huynh-Thu et al. [2010] proposent d'utiliser deux approches afin de construire ces ensembles d'arbre au sein de leur algorithme *GENIE*3. Le concept des *forêts aléatoires* représente la première d'entre-elles, dans ce cas, chaque arbre utilise un échantillon *bootstrap* des observations. De plus pour chaque nœud, la variable discriminante est choisie de manière optimale parmi un sous-ensemble aléatoire de k gènes possibles. La deuxième technique est celle des *Extras-Trees* où chaque arbre est construit à partir des mêmes observations initiales et où le choix du gène discriminant s'effectue

toujours parmi un sous ensemble aléatoire de k gènes. Cependant cette fois le seuil est également fixé aléatoirement pour chacun des k gènes discriminants possibles et non plus déterminé de manière optimale.

L'importance d'un gène discriminant pour un gène cible G_i est alors défini comme la somme des importances de ce gène sur l'ensemble des arbres construits pour G_i. Hormis le nombre d'arbres générés pour chaque gène, le second paramètre de $GENIE3$ est la valeur du paramètre k correspondant à la taille du sous-ensemble des gènes discriminants possibles. Diminuer la valeur de k permet d'augmenter le caractère stochastique de la méthode ce qui se révèle souvent bénéfique dans l'apprentissage de structure mais qui nécessite un nombre d'arbres plus important afin d'explorer suffisamment l'espace des arbres possibles. Huynh-Thu et al. [2010] proposent dans leur article d'utiliser $k = \sqrt{p-1}$ qui semble offrir un bon compromis.

6.2.2 Modèles graphiques gaussiens

Le calcul des corrélations partielles d'ordre $p-2$, bien que non calculable dans le cas général, est rendu possible sous la condition que les données suivent une loi gaussienne. On rentre alors dans le cadre des modèles paramétriques et plus précisément dans celui des *modèles graphiques gaussiens* (GGM). Le caractère graphique de ces modèles permet de représenter naturellement un RRG, chaque gène étant représenté par une variable du graphe dont les observations correspondent aux mesures d'expression et chaque arc dans ce graphe indique une dépendance conditionnelle entre deux variables sachant l'ensemble des $p-2$ autres variables. Ces méthodes basées sur la mesure des corrélations partielles ne permettent donc toujours pas de déduire une quelconque orientation des relations apprises.

Plus formellement considérons un ensemble de p variables observées $\mathbf{X} = \{X_1, X_2, .., X_p\}$ suivant une loi normale multivariée $\mathcal{N}_p = (\mu, \Sigma)$ où μ et Σ représentent respectivement les p moyennes et la matrice de covariance (de dimension $p \times p$) de $\mathbf{X}$. Lorsque Σ est inversible, c'est-à-dire qu'il existe une matrice inverse Σ^{-1} telle que $\Sigma\,\Sigma^{-1}$ correspond à la matrice unité, alors Σ^{-1} permet de calculer les coefficients partiels d'ordre $p-2$, noté ω_{ij}. On a ainsi pour chaque couple (X_i,X_j)

$$\omega_{ij} = \frac{\Sigma^{-1}_{ij}}{\sqrt{\Sigma^{-1}_{ii}\,\Sigma^{-1}_{jj}}}$$

Σ^{-1} est également appelée matrice de concentration ou encore matrice de précision. De manière similaire aux corrélations de Pearson, $\omega_{ij} = 0$ signifie que les variables X_i et X_j sont indépendantes sachant toutes les autres variables du modèle. Afin d'apprendre le RRG il suffit donc de calculer Σ^{-1} et de rechercher tous les coefficients ω_{ij} significativement différents de 0. Cependant dans le cas de données de grandes dimensions, telles que celles couramment rencontrées pour la reconstruction de RRG, l'estimateur empirique de Σ n'est généralement pas inversible ce qui empêche d'estimer les coefficients ω_{ij}.

Approches existantes

Corrélation d'ordre inférieur à p-2 Plusieurs voies ont été explorées afin d'estimer Σ^{-1}, la première d'entre elles consiste à revenir au cas de corrélations d'ordre limité de la même manière que le logiciel *ParCorA*. Ces approches utilisent le postulat selon lequel les RRG sont peu denses et que des ensembles de conditionnement de quelques gènes suffisent à détecter une grande majorité des indépendances. Ainsi Wille and Bühlmann [2006] effectuent pour chaque couple (G_i,G_j) le test du rapport de vraisemblance simple et conditionnelle à une seule variable $G_k \neq \{G_i,G_j\}$ entre les deux hypothèses H_0 et H_1 où $H_0 : \Sigma^{-1}_{ij} = 0$ et $H_1 : \Sigma^{-1}_{ij} \neq 0$. Les $p - value$ obtenues sont ensuite corrigées pour tenir compte de la multiplicité des tests. Si la valeur maximale de ces $p - value$ corrigées est supérieure à un seuil α, défini par l'utilisateur, alors l'arc entre G_i et G_j est ajouté au graphe.

De manière similaire Castelo and Roverato [2006] utilisent un ensemble de conditionnement de taille q où $1 \leq q \leq p - 2$. Au vu de l'augmentation exponentielle du nombre d'ensembles possibles à q variables de contrôle, seul un sous-ensemble des combinaisons est testé pour chaque couple (G_i,G_j). Le nombre de configurations parmi ce sous-ensemble où l'hypothèse H_0 est rejetée sous une seuil α donné permet alors de définir un taux de rejet. Seuls les couples dont le taux de rejet est supérieur à un second seuil β sont reliés dans le graphe. La taille de l'ensemble de conditionnement q est un paramètre important dont dépend le réseau reconstruit. La méthode proposée permet d'adapter ce nombre q durant la recherche, ce réglage est alors guidé à la fois par la valeur des différents taux de rejet et par les variations de connectivité des graphes produits en fonction de β.

Régressions indépendantes La seconde piste explorée utilise la relation entre les valeurs de Σ^{-1}_i non nulles associées à G_i et les coefficients issues de la régression de G_i sur l'ensemble des autres gènes. Meinshausen and Bühlmann [2006] proposent alors de définir pour chaque gène un modèle de régression linéaire de type

$$G_i = G\beta_i + \epsilon$$

où G_i est le vecteur des observations du gène G_i, G la matrice $n \times p$ de l'ensemble des n échantillons sur les p gènes servant à prédire G_i, β_i le vecteur des coefficients d'interaction de G_i avec les p gènes du réseau et ϵ un bruit blanc supposé gaussien. Résoudre ce modèle consiste à estimer les p paramètres β_{ij}, $j \in [1, p]$. Chaque paramètre β_{ij} reflète l'interaction entre les gènes G_i et G_j, une valeur nulle indiquant l'absence d'interaction. On notera que chaque paramètre β_{ii} qui correspond à une interaction de G_i avec lui même est fixé à 0 afin d'empêcher la régression d'un gène sur lui même.

Parmi les différentes techniques permettant de résoudre ce type de modèle, Meinshausen and Bühlmann [2006] utilisent la régression Lasso. L'estimation des paramètres β_i s'effectue alors en minimisant la somme des erreurs quadratiques sous une contrainte de type ℓ_1. On a ainsi :

$$\hat{\beta}_i(t) = \text{argmin}_{\beta \in \mathbb{R}_p} \{ \|G_i - G\beta_i\|_{\ell_2} \} \text{ sous contrainte } \| \beta_i \|_{\ell_1} \leq t$$

Cette régression est dite pénalisée du fait de la contrainte $\|\beta_i\|_{\ell_1} = \sum_j |\beta_{ij}| \leq t$. Le paramètre t représente alors le poids de cette pénalité, une valeur élevée de t équivaut à une estimation non

pénalisée au sens des moindres carrés tandis qu'une valeur plus faible limite le nombre de variables explicatives à inclure dans le modèle. La valeur du paramètre t peut être déterminée suivant un critère tel que *BIC* ou en utilisant une technique de validation croisée.

Cet estimateur s'exprime de manière équivalente comme

$$\hat{\beta}_i(t) = \text{argmin}_{\beta \in \mathbb{R}_p} \{ \|G_i - G\beta_i\|_{\ell_2} + \lambda \| \beta_i \|_{\ell_1} \}$$

où λ représente la pénalité de l'estimateur. Pour chaque valeur de t fixée nous pouvons trouver une valeur λ menant à la même estimation et réciproquement.

Le cadre des régressions pénalisées a donné lieu à de nombreuses extensions, soit sur la partie pénalisée, tel la norme ℓ_1 pondérée avec le *Lasso adapté* [Zou, 2006] ou la somme des normes ℓ_1 et ℓ_2 avec *ElasticNet* [Zou and Hastie, 2005], soit sur la partie du critère de vraisemblance avec par exemple le *square root Lasso* [Belloni et al., 2011].

De plus, les régressions pénalisées peuvent être vues comme les estimations MAP (*Maximum a posteriori*) de vraisemblances bayésiennes pour lesquelles le seuil de la pénalisation est une distribution de dirac. Dans ce cadre bayésien, une autre loi *a priori* sur le seuil de pénalisation a été proposée, conduisant au *lasso bayésien* [Hans, 2009].

Une fois les p régressions effectuées pour une valeur t fixée, un RRG non-orienté est obtenu en ajoutant un arc entre G_i et G_j dès lors que $\hat{\beta}_{ij} > 0$.

Plutôt que d'estimer directement les paramètres du modèle de régression d'un gène sur l'ensemble des autres gènes à l'aide d'une régression pénalisée, Verzelen [2010] propose de rechercher ce modèle en deux temps. La première étape consiste à estimer les paramètres de l'ensemble des modèles de régression possibles d'un gène sur un nombre maximum k de gènes. Pour chacun des $\sum_{l=1}^{k} C_{p-1}^{l}$ modèles possibles, les paramètres sont estimés afin de minimiser la somme des erreurs quadratiques sans pénalité. Puis l'un de ces modèles est retenu suivant un critère composé de l'erreur précédemment calculée et d'une pénalité dépendante du nombre de variables explicatives dans le modèle. A la différence de Meinshausen and Bühlmann [2006] cette pénalité ne dépend pas de Σ^{-1} ce qui lui assure une plus grande stabilité même dans le cas de la très grande dimension.

Nous pouvons également citer les travaux de Gardner et al. [2003] et de Wildenhain and Crampin [2006] développés dans le cas de données de perturbation. Chacune des p observations utilisées correspond alors au niveau d'expression stabilisé de l'ensemble des gènes après une perturbation ciblée sur l'un d'entre eux. Ces niveaux d'expression sont définis relativement à l'observation de référence pour laquelle aucune perturbation n'est appliquée. Chaque gène est modélisé par une équation linéaire différentielle dont le terme de la dérivée s'annule dans le cas de données stabilisées. Le problème se ramène alors à l'estimation des paramètres d'une équation linéaire classique. Gardner et al. [2003] proposent d'estimer pour un gène les paramètres de tout les modèles de régression possible à k gènes explicatifs en minimisant la somme des erreurs. L'algorithme sélectionne alors le modèle qui minimise au mieux cette erreur. Cette approche est similaire à la première étape de la procédure de Verzelen [2010] décrite précédemment. Au lieu de fixer un nombre de variables explicatives commun à chaque

gène Wildenhain and Crampin [2006] utilisent un score pénalisé, les gènes explicatifs sont alors ajoutés au modèle de manière itérative afin de minimiser le critère AIC.

Estimation de Σ^{-1} Plutôt que de décomposer le problème en p équations indépendantes, certains travaux visent à estimer au mieux Σ^{-1} sans pour autant se limiter aux corrélations d'ordre inférieur à $p-2$. Schäfer and Strimmer [2005] proposent ainsi dans leur logiciel *GeneNet* toute une stratégie dans ce but. La première étape consiste à calculer la pseudo-inverse de Σ, puis à utiliser une approche *bootstrap* afin de stabiliser les coefficients de cette pseudo-inverse. Puis chacun de ces coefficients est testé par rapport à l'hypothèse nulle (H_0 correspond ici à une corrélation égale à 0). Cependant, ces tests requièrent la distribution de H_0 qui ne peut être analytiquement obtenue dans le cas d'une pseudo inverse. Afin d'estimer cette distribution, les auteurs utilisent l'hypothèse de faible densité du RRG ce qui implique qu'une majorité des coefficients estimés vérifie H_0. La distribution de H_0 est alors estimée à partir de ces coefficients. Finalement le seuil de significativité sur les *p-value* de ces tests est défini afin de contrôler le FDR dans le cas de tests multiples.

D'autres approches visent à estimer Σ^{-1} par maximum de vraisemblance. Afin d'assurer une faible densité du graphe résultant une pénalité est assortie au modèle. Yuan and Lin [2007] et Friedman et al. [2008] utilisent ainsi une pénalité de type Lasso. Les premiers reformulent ce problème d'estimation en un problème d'optimisation convexe résolu à l'aide d'un algorithme de point intérieur tandis que les seconds utilisent une technique d'optimisation par bloc dans la matrice Σ. Cette dernière méthode proposée par Friedman et al. [2008] dite du *glasso*, équivaut à itérer jusqu'à convergence des régressions Lasso dont la valeur des coefficients tient compte des régressions précédentes. Les auteurs pointent alors un parallèle entre le *glasso* et la méthode des p régressions indépendantes proposée par Meinshausen and Bühlmann [2006]. La différence réside uniquement dans la conception itérative de *glasso* avec mise à jour des coefficients, là où Meinshausen and Bühlmann [2006] effectuent des régressions indépendantes une seule fois.

Cet algorithme *glasso* est utilisé par Chiquet et al. [2009] dans leur logiciel *SIMoNe*, qui offre la possibilité d'inclure un *a priori* sur une structure modulaire du graphe. Une fois le nombre de module fixé par l'utilisateur, l'algorithme itère deux phases du type EM. Lors de la première phase chaque gène est assigné à un module puis la seconde phase consiste à appliquer le *glasso* dont la pénalité est modifiée afin de prendre en compte l'*a priori* de modularité du réseau.

Nous pouvons également citer les travaux de Giraud et al. [2009] dont le logiciel *GGMselect* regroupe plusieurs des approches présentées précédemment afin de générer à partir de chacune d'elles des familles de graphes. Le logiciel retourne alors le graphe qui minimise un critère *Crit* similaire à celui employé par Verzelen [2010]. Quatre familles de graphes sont proposées, la famille "C01" correspond aux mesures de corrélation partielle d'ordre 0 et 1 telles que définies par Wille and Bühlmann [2006], la famille "LA" issue des régressions Lasso indépendantes proposées par Meinshausen and Bühlmann [2006], la famille "EW" représente les régressions Lasso adaptées et enfin la famille "QE" qui maximise localement le critère *Crit*. Les trois premières approches étant basées sur des régressions pénalisées, l'algorithme fait varier cette pénalité afin de générer les familles associées. Ces familles sont donc composées des graphes obtenus en faisant varier cette pénalité de

la valeur la plus stringeante jusqu'à une valeur limite correspondant à une borne sur le degré entrant maximal du graphe. La famille "QE" plus spécifique, est basée sur la construction des deux graphes G_{AND} et G_{OR} à partir des ensembles de voisins de chaque gène, définis localement suivant le critère *Crit*. Le graphe G_{AND} est composé des arcs reliant les paires de gènes dont chacun d'eux appartient à l'ensemble des parents du second. A l'inverse le graphe G_{OR} contient un arc entre deux gènes dès que l'un d'eux appartient au voisinage de l'autre. Ainsi la famille "QE" contient l'ensemble des graphes compris entre G_{AND} et G_{OR}, c'est-à-dire ceux qui incluent tous les arcs de G_{AND} sans inclure d'arcs n'appartenant pas à G_{OR}.

Approches bayésiennes Enfin une dernière possibilité consiste à approximer Σ^{-1} par une distribution de probabilité, ce qui permet de calculer la vraisemblance marginale d'un graphe dans un cadre bayésien. Dobra et al. [2004] et Scott and Carvalho [2008] utilisent la distribution de Wishart dans ce sens. Un score bayésien est alors défini comme le produit de la vraisemblance marginale du graphe et d'une pénalité sur sa densité. Dobra et al. [2004] proposent de sélectionner pour chaque variable un ensemble de variables explicatives restreint par un ordre de la même façon que l'algorithme K2 présenté dans la section 2.1.2.3. A la différence de K2, l'ordre est construit progressivement durant la recherche. A chaque itération la variable la moins corrélée au reste du réseau est sélectionnée, suivie par l'estimation de ses prédicteurs avant d'être retirée du réseau. Au terme de p itérations les ensembles de prédicteurs sélectionnés pour chaque variable permettent de construire un DAG dont l'orientation respecte l'ordre produit. La sélection des variables explicatives s'effectue par une méthode de type *forward-backward* guidée par le score bayésien.

Plutôt que de rechercher pour chaque variable l'ensemble de ses parents, Scott and Carvalho [2008] proposent de parcourir l'espace des graphes décomposables non-orientés à l'aide d'une procédure de type MCMC guidée par le score bayésien. La recherche s'effectue via des déplacements locaux lors d'ajout ou de suppression d'arcs mais aussi de modification plus globales du graphe afin d'échapper aux situations d'optimum local. Cette méthode qui retourne une probabilité *a posteriori* de chaque arc en fonction des graphes visités et de leur vraisemblance marginale, permet de reconstruire par exemple le graphe basé sur les arcs de maximum *a posteriori* (MAP).

La majorité des méthodes développées dans le cadre des modèles graphiques gaussiens définissent des graphes non-orientés dû au caractère symétrique des mesures de corrélation. Ce graphe ne permet donc pas d'en déduire directement un RRG. Afin d'orienter ces arêtes l'une des possibilité consiste à perturber expérimentalement certains gènes du réseau et à observer l'effet produit sur les autres gènes. Par ailleurs nous avons vu dans la première partie de ce manuscrit que les RB orientent certaines relations grâce aux *v-structures* rendant ce formalisme attractif pour la reconstruction de RRG. Nous présentons donc par la suite quelques approches utilisant les RB.

6.2.3 Réseaux bayésiens

Nous avons déjà détaillé dans la première partie de ce manuscrit les différentes méthodes développées afin d'apprendre la structure d'un RB. Nous donnons ici quelques exemples appliqués à des données d'expression.

L'une des approches les plus connues dans ce domaine étant celle de Friedman et al. [2000] qui tente de répondre aux deux difficultés majeures de l'apprentissage des RRG. La première est liée au nombre élevé de variables du réseau engendrant de fait un vaste espace de recherche tandis que la seconde difficulté provient du faible nombre d'échantillons disponibles diminuant la robustesse des arcs appris. Afin de réduire l'espace de recherche les auteurs utilisent l'algorithme *Sparse Candidate*. Ce dernier est par ailleurs modifié via une mise à jour de la liste restreinte des k parents prometteurs après chaque modification du graphe et non pas au terme de chaque recherche gloutonne complète. Puis une technique de *bootstrap* est employée afin d'augmenter la robustesse des prédictions.

Afin de réduire l'espace de recherche Edwards et al. [2010] proposent d'utiliser l'algorithme de Chow and Liu [1968] muni d'un critère pénalisé de type *BIC*, restreignant alors l'apprentissage à l'espace des arbres ou plus précisément celui des forêts. En effet l'utilisation d'un critère pénalisé engendre des pondérations négatives, ainsi chaque arc ayant une telle pondération n'est alors pas considéré pour la construction de l'arbre, ce qui n'assure plus la construction d'un arbre couvrant. Ces mêmes auteurs proposent également d'apprendre des modèles mixtes composés de variables discrètes et continues.

Zhu and Li [2010] effectuent quant à eux une recherche contrainte par couche. Les variables d'expression sont classées d'après la littérature en différentes couches, puis l'ensemble optimal de parents pour chaque variable est sélectionné parmi les variables présentes dans la couche supérieure. Cette idée similaire à l'algorithme K2, ne nécessite cependant pas un classement aussi fin, les auteurs utilisent ainsi 4 couches afin de reconstruire une voie de signalisation.

On remarque d'après ces quelques exemples que l'application des RB à la dimension des RRG requiert bien souvent une restriction de l'espace de recherche, justifiant ainsi l'utilisation d'un filtre tel que présenté en section 4.2.

D'autres travaux utilisent des approches évolutionnaires comme ceux de Ram and Chetty [2007], Auliac et al. [2008] où l'évolution de la population des DAG s'effectue par croisement entre paires de DAG durant lequel apparaît des phénomènes de recombinaison. A la suite de chaque génération, afin de maintenir la taille de la population constante, une fonction de score sélectionne les DAG à conserver parmi l'ensemble des DAG courants. Afin d'explorer plus largement l'espace des DAG et éviter une convergence trop rapide de la population vers un ensemble de DAG similaires, un système de mutation spontanée est également utilisé. Le ratio entre la diversité de la population et sa convergence vers des structures à haut score est un aspect important dans ce type d'approche. Ram and Chetty [2007] proposent ainsi d'alterner des phases de diversification et de concentration de la population afin de maintenir un niveau de diversité moyen au fil des générations. Auliac et al. [2008] proposent quant à eux une nouvelle stratégie de sélection des individus à conserver dans la population. Ainsi, au lieu de

ne conserver que les DAG maximisant la fonction de score, il est proposé qu'un DAG fils remplace l'un de ses DAG parents dans la population, si celui-ci possède un score élevé. Autrement dit, un seul représentant est conservé parmi ces deux graphes similaires, évitant ainsi une convergence trop rapide de la population.

Outre l'utilisation d'approches à base de score, d'autres méthodes recherchant directement les indépendances ont été proposées pour l'apprentissage de RRG, Tan et al. [2011] décrivent ainsi deux extensions à l'algorithme PC. La première vise à considérer l'apprentissage de réseaux peu denses mais dont certaines variables possèdent une forte connectivité, cette structure étant typique des RRG connus. La deuxième consiste à fixer la stringence du test d'indépendance suivant un *a priori* sur les relations entre variables, cet *a priori* pouvant provenir par exemple de la littérature existante.

Cette idée d'intégrer d'autres sources de données est justement détaillée dans la section suivante où nous nous focaliserons spécifiquement sur l'utilisation de données de polymorphisme en plus des niveaux d'expression.

6.3 Approches intégratives

Ces dernières années de nouvelles techniques dites *intégratives* utilisent en plus des classiques données d'expression diverses sources d'information. Ainsi Tamada et al. [2003] développent une méthode basée sur les RB prenant en compte l'information des sites de fixation des facteurs de transcription. Un premier réseau est d'abord reconstruit uniquement à partir des données d'expression. Puis la recherche de motifs communs présents dans les régions promotrices des gènes co-régulés d'après ce réseau, amène à définir des facteurs de transcription potentiels pour ces gènes. Cette information sert alors d'*a priori* lors d'une nouvelle phase d'apprentissage du réseau.

A partir de données de génétique-génomique D'autres méthodes utilisent l'information issue des polymorphismes entre individus. Comme nous l'avons décrit précédemment, ces mutations naturelles provoquent des variations dans l'expression des gènes, des relations de causalité peuvent donc être tracées des polymorphismes vers les gènes. La détection de ces polymorphismes causaux s'effectue généralement via une recherche d'eQTLs, on obtient ainsi pour chaque gène d'intérêt, des régions du génome dont les variations génétiques semblent expliquer le niveau d'expression de ce gène. Jansen and Nap [2001] introduisent alors la notion de données de *génétique-génomique* et expliquent le processus permettant de déduire de ces eQTLs, des régulations possibles entre gènes. Afin d'illustrer ce processus, prenons l'exemple de la Figure **6.2** tirée de Jansen and Nap [2001].

La détection des eQTLs s'effectue à l'aide de marqueurs génétiques, ces marqueurs permettent de lire un ou plusieurs nucléotides sur la séquence et donc de repérer des situations de polymorphisme. Les marqueurs jouent alors le rôle de balises couvrant de manière plus ou moins fine l'intégralité du génome afin de détecter les eQTLs. Une région eQTL s'étend généralement sur plusieurs marqueurs, il est toutefois possible de restreindre cette région au marqueur ayant le score LOD le plus élevé. On obtient ainsi pour chaque gène un unique marqueur par eQTL détecté, sur la Figure **6.2** cette information est représentée sous la forme de la matrice (a). La carte génétique (b) indique la

localisation sur les chromosomes des gènes (en chiffre) et des marqueurs (en lettre), certains gènes peuvent alors être co-localisés avec un marqueur mais la bijection n'est pas obligatoire, ainsi aucun marqueur n'est situé au niveau du gène 6. Il est possible de construire un RRG probable (c) à partir de ces deux seules informations, en remplaçant dans la matrice (a) chaque marqueur par le gène situé à la même position (si celui-ci existe). Prenons ainsi l'exemple du gène 4 qui possède d'après la matrice (a), un eQTL au niveau de chacun des marqueurs D et F. La carte génétique nous indique que le marqueur D est justement situé au niveau du gène 4. Dans cette situation l'eQTL détecté indique seulement l'existence d'un effet *cis* du polymorphisme sur le gène 4. Le marqueur F est quant à lui situé à proximité du gène 3, indiquant un effet *cis-trans* ou *trans* de ce marqueur sur le gène 4. On en déduit alors une régulation orientée du gène 3 vers le gène 4 dans le réseau (c). On remarquera par ailleurs qu'aucun eQTL n'explique le gène 3, l'effet *cis* du marqueur F sur ce gène est donc peu probable ce qui mène à préférer l'existence d'un effet *trans* du marqueur F sur le gène 4.

Bien que ce réseau puisse être construit sans le calcul des corrélations entre niveaux d'expression, celui-ci suppose que toutes les variations d'expression puissent être expliquées au travers des mutations. Il est donc peu probable d'utiliser ce réseau tel quel, cependant il représente une importante source d'information utilisable par les méthodes classiques de reconstruction de RRG. Rockman [2008] effectue une synthèse de l'utilisation de ce type de données en présentant leurs intérêts pour la reconstruction de RRG.

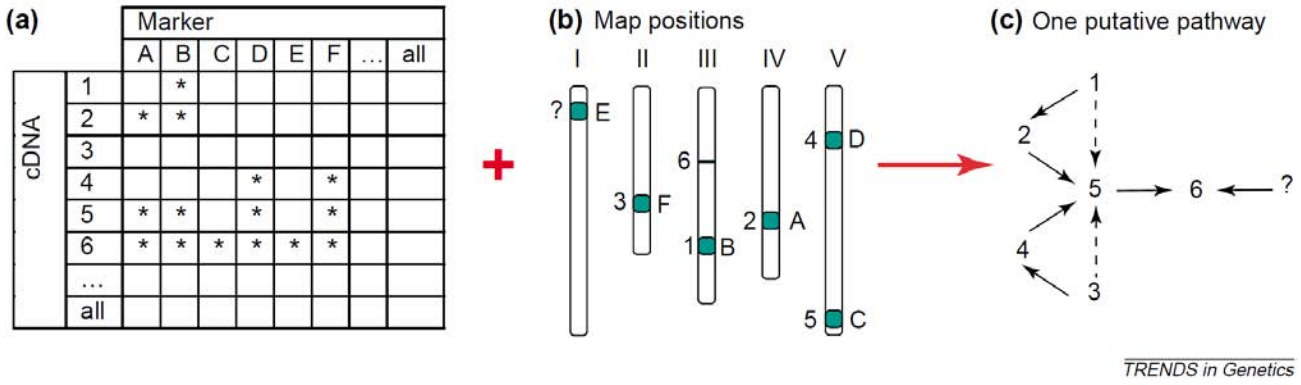


Figure 6.2 – **(a)** Matrice indiquant pour chaque gène (en ligne) la détection d'une région eQTL centrée sur un marqueur (en colonne). **(b)** Position des gènes et des marqueurs sur la carte génétique. **(c)** Réseau de régulation possible obtenu par projection des relations de la matrice (a) en remplaçant chaque marqueur avec le gène co-localisé d'après la carte génétique (b).

Méthodes employées La première utilisation des eQTLs n'a cependant pas eu pour objectif de reconstruire des régulations entre gènes. Schadt et al. [2005] proposent ainsi d'utiliser les eQTLs afin de trouver les gènes expliquant un trait d'intérêt, à savoir l'obésité chez la souris. La première étape consiste à sélectionner chaque gène pour lequel un eQTL a été détecté et dont la zone de ce dernier

chevauche la région génomique reconnue comme étant responsable de l'obésité. Puis pour chacun d'eux, trois situations possibles sont testées afin de distinguer si le gène est dans une situation causale, réactive ou indépendante par rapport au trait d'intérêt sachant l'eQTL commun. En d'autres termes, ce test détermine si le gène explique le trait ou bien si le gène est expliqué par le trait ou encore si le gène et le trait sont indépendants. Les gènes possédant un effet causal fort d'après ce test sont alors validés expérimentalement.

Ces mêmes auteurs proposent également de considérer comme trait d'intérêt l'expression d'un autre gène. Dans ce cas un gène est sélectionné dès lors qu'il existe un chevauchement d'un de ses eQTLs avec l'eQTL du gène d'intérêt. Les trois mêmes configurations sont ensuite testées afin de détecter la nature de la relation sachant l'eQTL commun. Ce test permet cette fois de définir des relations causales entre gènes, cependant cette approche ne permet pas de détecter toutes les régulations.

Zhu et al. [2007] reprennent alors cette idée afin de définir un *a priori* lors de la reconstruction globale d'un RRG artificiel dans le cadre des RB. Les couples de gènes dont les eQTLs se chevauchent sont testés pour les trois mêmes configurations. Les situations causales et réactives donnent alors lieu à un *a priori* positif sur l'existence d'un arc dans le réseau tandis que la situation d'indépendance diminue cette même probabilité *a priori*. Le réseau est ensuite reconstruit à partir des seules données d'expression en utilisant l'algorithme MCMC qui intègre l'*a priori* dans sa probabilité de transition. Zhu et al. [2008] poursuivent dans cette voie en incluant un second *a priori* issu de la correspondance entre la séquence des sites de fixation et celle des facteurs de transcription, appliqué au cas d'une levure.

Sans chercher de recouvrements d'eQTLs, Keurentjes et al. [2007] utilisent plus simplement ces régions pour identifier dans celles-ci les régulateurs potentiels de gènes d'intérêt impliqués dans le processus de floraison d'*Arabidopsis thaliana*. Puis parmi ces régulateurs potentiels, les gènes les plus corrélés avec les gènes d'intérêt sont utilisés pour construire le réseau de régulation.

Ou encore Logsdon and Mezey [2010] qui s'intéressent spécifiquement aux gènes pour lesquels un eQTL de type *cis* a été détecté. Pour ces gènes, une régression dérivée du Lasso est utilisée afin de sélectionner leurs variables explicatives parmi lesquelles est inclus le marqueur correspondant à l'eQTL détecté. La reconstruction du réseau est ainsi restreinte aux seuls gènes soumis à un effet *cis* et aux marqueurs correspondant à ces eQTLs détectés. Ces mêmes auteurs proposent plus récemment [Logsdon et al., 2012] d'utiliser une régression pénalisée via un *a priori* bayésien sur les paramètres, afin de sélectionner les gènes et marqueurs explicatifs de différents traits d'intérêts chez la souris.

L'information de ces eQTLs peut également être utilisée à des fins d'orientation. Chaibub Neto et al. [2008] proposent ainsi d'orienter les arêtes d'un réseau initial suivant les eQTLs détectés. Ce réseau initial pouvant être obtenu par des approches classiques d'apprentissage de réseau, les auteurs proposent ainsi d'utiliser la première phase de l'algorithme PC. Un premier test basé sur un score pénalisé est d'abord réalisé afin de distinguer les eQTLs ayant un effet direct sur le gène. Puis les arêtes sont orientées suivant le rapport entre la vraisemblance des deux orientations, conditionnellement à ces eQTLs directs.

Plutôt que d'effectuer une recherche d'eQTL préalable, d'autres approches considèrent directement les marqueurs comme variables du modèle.

Chu et al. [2009] reprennent ainsi l'approche de Schäfer and Strimmer [2005] afin de construire dans un premier temps un RRG à partir des seuls niveaux d'expression. Un sous-réseau est ensuite défini à partir des relations les plus significatives. Les marqueurs situés à proximité des gènes composant ce sous-réseau sont alors introduits à leur tour dans le modèle. Le sous-réseau ainsi enrichi est appris une seconde fois afin de détecter les effets *cis* potentiels.

Chipman and Singh [2011] considèrent également dans leur méthode *SCT* les marqueurs comme des variables du modèle. L'apprentissage se déroule en deux étapes. La première consiste à considérer chacun des p marqueurs comme racine d'une arborescence construite de manière itérative par l'ajout successif de variables d'expression. A chaque itération une nouvelle variable est reliée à l'un des nœuds constituant l'arborescence courante d'après un critère stochastique guidé par la vraisemblance de l'ajout. L'arborescence croit de cette manière jusqu'à atteindre une taille fixée. Pour chacun des p marqueurs ce processus est répété r fois afin de construire une forêt composée des $p \times r$ arborescences. Un *a priori* est alors calculé pour chaque arc reliant deux gènes suivant leur fréquence d'apparition dans cette forêt. Cet *a priori* est intégré à un algorithme MCMC qui constitue la deuxième phase de la méthode, la recherche s'effectue cette fois dans le cadre des RB où seules les variables d'expression sont considérées. A terme un classement des régulations est établi suivant leur fréquence d'apparition parmi les graphes rencontrés durant la recherche.

6.4 Conclusion

Nous venons de voir dans ce chapitre, les principales approches permettant d'apprendre un RRG à partir de mesures d'expression statiques. Ces méthodes diverses utilisent majoritairement des données continues, parmi elles les modèles graphiques gaussiens représentent une large part des méthodes proposées actuellement dans la littérature. D'autres méthodes telles que les arbres de régression représentent cependant un concept séduisant de par leur simplicité et leur efficacité en pratique. Les réseaux bayésiens discrets offrent quant à eux une alternative aux modèles linéaires, bien que nécessitant une phase de discrétisation, la modélisation d'effets non-linéaires reste un atout de poids dans le cadre de la reconstruction de RRG.

Nous avons également présentées quelques méthodes intégratives et plus particulièrement celles utilisant des données de *génétique-génomique*. Ces informations supplémentaires peuvent être utilisées de manière variée allant de la simple restriction de l'espace de recherche, à l'utilisation en tant que variables du modèle en passant par une utilisation comme *a priori* pour la reconstruction. Nous proposons dans le chapitre suivant une utilisation de ces données en tant que variables à part entière d'un réseau bayésien.

Chapitre 7

Comparaisons des approches

Sommaire

7.1 Modélisations proposées pour les réseaux bayésiens 136
7.1.1 Modèle non-fusionné 137
7.1.2 Modèle fusionné 140
7.2 Comparaison des approches 142
7.2.1 Comparaisons sur les réseaux Web-50 144
7.2.1.1 Réseaux de régulations utilisés 144
7.2.1.2 Génération des données de génétique-génomique 145
7.2.1.3 Critères d'évaluation 147
7.2.1.4 Méthodes comparées 147
7.2.1.5 Résultats 147
7.2.2 Compétition DREAM 155
7.2.2.1 Présentation de la compétition DREAM 155
7.2.2.2 Données du challenge 155
7.2.2.3 Critères d'évaluation 157
7.2.2.4 Méthodes comparées 158
7.2.2.5 Résultats officiels 161
7.2.2.6 Résultats supplémentaires 168
7.3 Conclusion 172

Après avoir présenté les différentes approches utilisées afin de reconstruire des RRG, nous nous intéressons plus particulièrement à l'utilisation des réseaux Bayésiens. Nous commençons par présenter deux modélisations dans le cas des données de *génétique-génomique* que nous comparons dans un premier temps entre-elles uniquement avant d'élargir cette comparaison à d'autres approches de reconstruction de RRG. Nous analysons également l'effet de plusieurs propositions d'amélioration telles que l'utilisation de scores étendus présentés en section 4.3 ou la prise en compte de connaissances biologiques. Nous présentons par la suite les résultats obtenus pour l'un des challenges de la compétition internationale DREAM5 où l'objectif fût d'apprendre la structure d'un RRG à l'aide

de données de *génétique-génomique* et pour laquelle l'équipe SaAB a proposé différentes approches dont l'une basée sur les RB.

7.1 Modélisations proposées pour les réseaux bayésiens

Les réseaux bayésiens en tant que modèles graphiques permettent de modéliser aisément les réseaux de régulation de gènes. Chaque variable du RB représente un gène et chaque arc du graphe, une régulation entre deux gènes. La mesure des p niveaux de transcrit pour une population constituée de n individus représente ainsi les observations pour chacune des p variables du RB notées $G_i,\ i \in [1, p]$.

Nous avons vu précédemment que l'information des polymorphismes peut être utilisée afin de fournir un *a priori* sur la structure du réseau [Zhu et al., 2007] ou en tant que variables à part entière [Chipman and Singh, 2011]. Nous choisissons cette dernière solution en utilisant des variables marqueurs informant de ces polymorphismes. Dans notre cas nous supposerons l'utilisation de marqueurs bi-allélique qui comme son nom le laisse supposer, permet de différencier deux allèles d'un même gène.

Le terme mutation utilisé jusque ici nécessite cependant quelques précisions avant de poursuivre, la présence d'une mutation sous entend l'existence d'une séquence de référence. On supposera donc par la suite que l'un des deux individus fondateurs à la population contrôlée (P1 ou P2 sur la Figure **6.1**) représente les allèles de référence tandis que le second représente les allèles mutants. La présence d'une mutation observée à l'aide d'un marqueur pour un individu indique alors que celui-ci a hérité du nucléotide représentant l'allèle mutant tandis que l'absence de mutation sous entend que le nucléotide représente l'allèle de référence que nous supposons être différents. Nous utiliserons également parfois le terme de génotype afin de décrire l'état muté ou non d'un nucléotide, de même génotyper un individu revient à lire l'ensemble les génotypes au travers des différents marqueurs situés sur ses chromosomes.

Nous distinguerons donc par la suite les variables d'expressions G et les variables binaires de marqueur M permettant d'indiquer l'allèle observé (de référence ou mutant). Cette information supplémentaire ajoute donc des liens de causalité très fort car orienté, en effet si un polymorphisme provoque certaines variations d'expression, la réciproque n'est pas vraie. Les relations de régulations sont donc orientées d'un marqueur M_i vers un gène G_j. Lorsqu'il est possible de positionner sur la séquence ce marqueur M_i au niveau de la région codante d'un gène G_i alors la régulation apprise $M_i \rightarrow G_j$ est de type *trans*. A l'inverse si M_i est situé au niveau de la région promotrice de G_i alors la régulation apprise $M_i \rightarrow G_j$ représente un effet *cis-trans*. Dans les deux situations l'arc appris fourni un *a priori* très fort sur la régulation $G_i \rightarrow G_j$.

Nous présentons donc dans cette section deux modélisations en représentant de manière explicite dans le réseau l'observation des allèles (ou génotypage) à l'aide de marqueurs génétiques. La première d'entre-elles distingue clairement les variables représentant l'expression des gènes de celles associées aux marqueurs, nous serons alors dans le modèle non-fusionné, tandis que la seconde modélisation, dite fusionnée, rassemble au sein d'une même variable l'expression d'un gène et le marqueur situé à

proximité immédiate de celui-ci.

Par ailleurs nous supposerons qu'il existe pour chaque gène, un unique marqueur associé qui indique la présence d'un polymorphisme au niveau de ce gène. De même nous considérons que ce polymorphisme (ou la mutation) possède obligatoirement un effet sur la régulation ou sur le produit de l'expression du gène au travers d'un effet *cis* ou d'effets *trans* sur les autres gènes régulés. Cette distinction s'effectue suivant la position du marqueur sur une région promotrice ou codante du gène associé. Nous omettons également le cas des polymorphismes silencieux, chaque polymorphisme possède donc un effet mesurable. Cette dernière hypothèse, bien que contraignante, n'est cependant pas nécessaire à la validité des deux modélisations proposées, comme nous le préciserons pour chacune d'elles.

Nous définissons pour chaque gène, en plus de la variable G_i indiquant son niveau d'expression, une seconde variable binaire $M_i,\ i \in [1, p]$ de même indice correspondant au marqueur indiquant la présence ou non d'une mutation d'un nucléotide à proximité immédiate du gène. La première modélisation proposée distingue clairement ces deux types de variables.

7.1.1 Modèle non-fusionné

Le modèle non-fusionné est constitué de $2p$ variables correspondant aux p variables d'expression et aux p variables marqueurs. La Figure **7.1** représente une modélisation non-fusionnée d'un réseau de régulation comportant 3 gènes.

Nous pouvons ainsi classer les arcs en 3 types :

Arc entre deux marqueurs Les marqueurs situés sur un même chromosome sont représentés sous la forme d'une chaîne de Markov d'ordre 1, cette relation indique que l'état d'un marqueur dépend uniquement du marqueur le précédent directement sur le chromosome. Cette hypothèse découle du fait que durant la reproduction d'organismes diploïdes, le phénomène de *crossing-over* mène à ce que chaque chromosome hérité ne soit pas la copie d'un unique chromosome mais plutôt la combinaison des deux chromosomes d'un même parent. La synthèse de ce chromosome fils débute par la lecture de la séquence d'un des deux chromosomes du père, puis lors d'un *crossing-over*, les rôles s'inversent et c'est le second chromosome paternel qui est lu afin de poursuivre la synthèse. Plusieurs *crossing-over* peuvent apparaître pour un même chromosome. Dans le cas des croisements de type RIL, les chromosomes des individus obtenus sont alors constitués d'une alternance de séquences de référence et de séquences mutantes. La chaîne des marqueurs représente donc la probabilité qu'un nucléotide provienne de l'un des deux allèles sachant l'allèle d'origine du marqueur le précédent sur la séquence. Cette probabilité varie en fonction des distances entre marqueurs, lorsque cette distance est faible la probabilité d'observer un *crossing-over* est également faible ce qui augmente la probabilité d'observer deux nucléotides d'un même allèle pour ces deux marqueurs, la liaison génétique est alors élevée. A l'inverse l'éloignement des deux marqueurs sur un même chromosome, augmente leur degré d'indépendance, la liaison génétique devient alors négligeable. Plutôt que d'utiliser la distance physique exprimée en nombre de paires de base, on définit cet éloignement en terme de distance

génétique. Ces distances génétiques le long du chromosome sont liées à la probabilité d'occurrence d'un *crossing-over* sur les distances courtes. On fait l'approximation qu'1 centiMorgan (noté cM) est égal à une probabilité d'occurrence d'un *crossing-over* de 1%.

Arc d'un marqueur vers un gène La position du marqueur par rapport au gène associé mène à deux types d'arc liant les marqueurs aux gènes. Lorsque le marqueur est situé dans la région promotrice du gène associé tel que le gène G_3 de la Figure **7.1**, l'arc $M_3 \rightarrow G_3$ représente l'effet *cis* du polymorphisme qui agit directement sur l'expression du gène. Lorsque le marqueur est situé dans la région codante du gène tel que G_2 (de même pour G_1) on obtient des arcs de type *trans* ($M_2 \rightarrow G_1$ et $M_2 \rightarrow G_3$) qui sont des projections des régulations de G_2 vers les autres gènes du réseau. Ainsi les deux régulations provenant de G_2 vers G_1 et G_3 sont projetées à partir de M_2 vers ces mêmes gènes. Ces arcs modélisent l'effet de la mutation sur la structure de la protéine générée par G_2 et non sur sa quantité dans la cellule. Ainsi la régulation de G_2 sur G_1 et G_3 dépend à la fois du niveau d'expression de G_2 et de la forme de la protéine synthétisée, représentée par l'état de M_2.

Arc entre gènes Les arcs entre gènes représentent le RRG classiquement appris à partir des données d'expression seules. Ces régulations constituent le but final de la reconstruction, les autres arcs permettent uniquement d'améliorer la qualité descriptive du modèle et les performances de l'apprentissage.

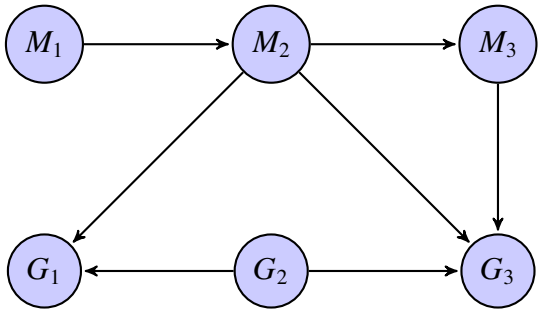


Figure 7.1 – Exemple d'un réseau de régulation entre 3 gènes : G_1, G_2 et G_3, chacun d'eux étant associé au marqueur de même indice M_i. Le gène G_2 régule les gènes G_1 et G_3 ($G_2 \rightarrow G_1$ & $G_2 \rightarrow G_3$). Les marqueurs génétiques sont liés suivant leur succession sur le chromosome de telle sorte que $M_i \rightarrow M_{i+1}$. Nous pouvons distinguer deux positions pour un marqueur. Si celui-ci se situe dans la région promotrice de son gène (cas de M_3), une mutation modifiera le niveau d'expression du gène, nous aurons ainsi $M_3 \rightarrow G_3$. Dans le cas où le marqueur est situé dans la région codante du gène (cas de M_1 et M_2) seule la force de régulation de ce gène sur les autres gènes sera modifiée lors d'une mutation et non son niveau d'expression. Dans ce cas pour chaque relation $G_i \rightarrow G_j$ (dans notre exemple $i = 2$ et $j \in \{1, 3\}$) nous aurons aussi $M_i \rightarrow G_j$.

Le principal inconvénient de cette modélisation réside dans le nombre doublé de variables par rapport au nombre de gènes observés, il est toutefois possible de réduire l'espace des graphes à considérer grâce à certaines connaissances biologiques ainsi qu'à l'aide des données de marqueurs.

Restriction des graphes possibles

Restrictions biologiques La Figure **7.2** montre les premières restrictions effectuées grâce aux principes de base de la génétique. Ces restrictions permettent de réduire les relations entre marqueurs à une simple chaîne et empêchent toute régulation d'un marqueur par l'expression d'un gène. On notera ainsi que le rôle des marqueurs dans le réseau est restreint à celui de parent des variables d'expression ce qui réduit fortement l'espace de recherche. L'apprentissage du RB se résume alors à rechercher l'ensemble des parents pour chacun des p gènes parmi les $2p-1$ variables possibles.

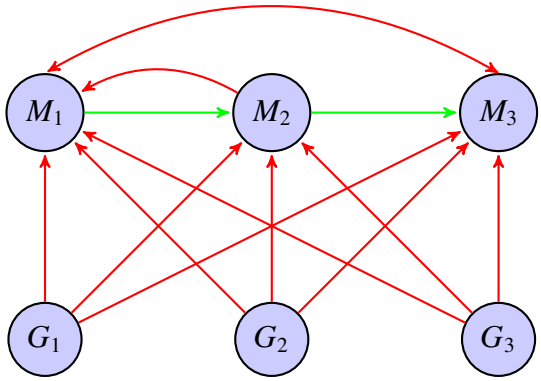


Figure 7.2 – Connaissant la position des marqueurs sur le chromosome, ceux-ci se modélisent sous la forme d'une chaîne de Markov d'ordre 1 suivant l'ordre $M_i \rightarrow M_{i+1}$ (en vert) tout en interdisant les autres relations entre marqueurs $M_i \nrightarrow M_j$, $\forall j \neq i+1$ (en rouge). Nous savons également qu'aucune expression de gène ne peut causer l'apparition d'une mutation $G_i \nrightarrow M_j$, $\forall i, j$ (en rouge).

Information des marqueurs Nous avons vu précédemment qu'il est possible de distinguer l'effet *cis* ou *trans* d'un SNP sur l'expression d'un gène à l'aide d'une analyse d'eQTL. Ces SNPs étant observés à l'aide de marqueurs situés au niveau de chaque gène, il est donc possible de classer nos p marqueurs suivant leur type d'effet. Par abus de langage nous définirons un marqueur comme étant de type *cis* si l'analyse eQTL indique qu'il existe pour le gène associé une région explicative chevauchant sa position sur le génome. Dans le cas inverse le marqueur est considéré comme *trans*. Nous noterons que cet effet *trans* s'exprime uniquement lorsque le gène associé régule d'autres gènes, ces arcs ne sont donc pas connus à l'avance mais permettent, si ils sont présents dans le réseau appris, de déduire des régulations entre gènes.

Nous pouvons donc imposer certaines restrictions à partir du type des marqueurs comme présenté sur la Figure **7.3**. Lorsque le marqueur est en *cis*, nous imposons un arc direct du marqueur vers le

gène modélisant ainsi cet effet. Nous supposons également dans ce cas que ce marqueur ne peux avoir d'effet direct sur l'expression d'un autre gène mais uniquement via l'expression du gène associé, nous interdisons donc les arcs de ce marqueur vers les autres gènes.

Lorsque le marqueur n'est pas en *cis*, nous avons vu que les arcs en *trans* issus de ce marqueur, sont des projections des régulations du gène associé vers les autres gènes, il est donc impossible de fixer de tels arcs sans connaître *a priori* ces régulations entre gènes. On ne peut alors qu'interdire l'arc direct reliant ce marqueur au gène associé dès lors que celui-ci représente une contradiction avec l'analyse eQTL qui n'a pas détecté d'effet *cis*.

Considérons un instant la présence d'un polymorphisme silencieux, d'après notre approche celui-ci ne pourra être détecté comme ayant un effet *cis* sur son gène, le marqueur sera donc traité, par défaut, comme ayant un effet *trans*. Dans ce cas aucun arc issu de ce marqueur n'est imposé, de plus l'absence d'effet au niveau de la régulation isolera cette variable du reste du réseau et n'impactera donc pas la reconstruction.

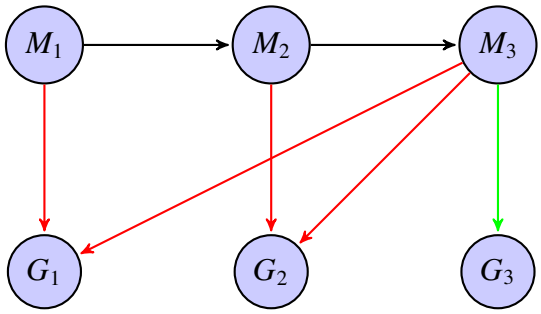


Figure 7.3 – Pour chaque marqueur M_i situé dans la région promotrice de son gène (cas de M_3) nous pouvons fixer l'arc $M_i \rightarrow G_i$ (en vert) et interdire les arcs $M_i \nrightarrow G_j$, $\forall j \neq i$ (en rouge). Dans le cas d'un marqueur M_i se situant dans la région codante (cas de M_1 et M_2) nous interdisons uniquement l'arc $M_i \nrightarrow G_i$ (en rouge).

Ces restrictions réduisent pour chaque gène le nombre de marqueurs possibles à considérer comme parent, seuls les marqueurs de type *trans* peuvent être des parents potentiels pour les variables d'expressions, les marqueurs *cis* étant quant à eux, liés exclusivement à leur gène associé. Le nombre de RB possibles est alors de $2^{p\ m_{cod}} \times N$ avec m_{cod} le nombre de gènes dont le marqueur se situe en région codante et N le nombre de DAG possibles pour les p variables d'expression.

7.1.2 Modèle fusionné

Une modélisation alternative basée sur la fusion des deux variables associées à un gène (M_i et G_i) permet cette fois de ne pas augmenter le nombre de variables du modèle. De cette fusion résulte une nouvelle variable E_i dont le domaine sera le produit cartésien de celui de M_i et G_i comme décrit sur la Figure **7.4**

Cette modélisation permet donc de représenter de manière compacte les deux sources d'information. Cependant dans ce modèle chacun des arcs $E_i \rightarrow E_j$ regroupe 4 arcs distincts du

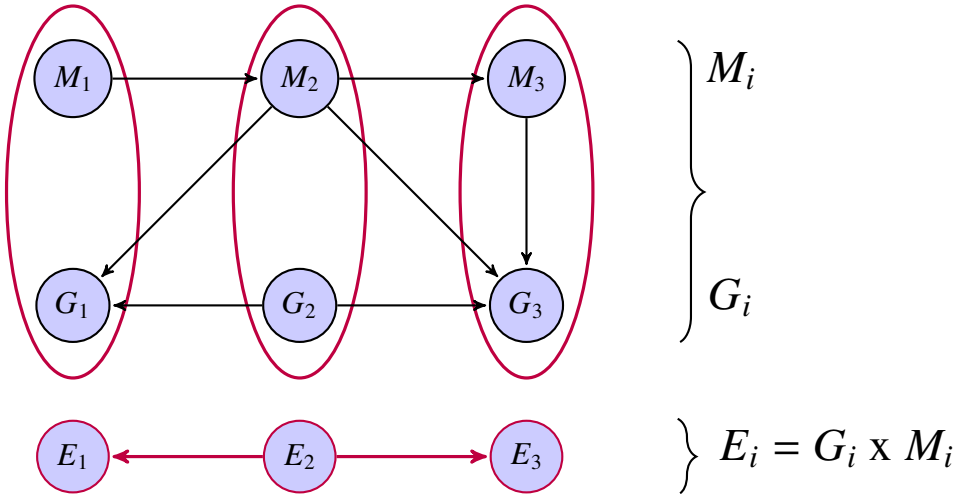


Figure 7.4 – Fusion du réseau de régulation comportant 3 gènes et 3 marqueurs en un modèle ne comportant que 3 variables E_i.

précédent modèle et de différentes natures ($\{M_i, G_i\} \rightarrow \{M_j, G_j\}$). L'une de ces 4 relations correspond notamment à une relation entre deux marqueurs, qui, dans le cas de marqueurs situés sur un même chromosome, existe indépendamment des régulations entre gènes. La situation est d'autant plus délicate que les marqueurs sont proches, la liaison génétique étant plus importante dans ce cas. Nous risquerions donc d'apprendre une relation entre deux variables $E_i \rightarrow E_j$ due uniquement au lien génétique $M_i \rightarrow M_j$. Afin d'éviter d'apprendre ces relations, il convient de corriger les fonctions de score pour ce modèle.

Le calcul de la log-vraisemblance des données $\log(\mathbb{P}(\mathbf{D}|\mathcal{G}))$ peut ainsi s'écrire comme suit :

$$\begin{aligned}
\log(\mathbb{P}(\mathbf{D}|\mathcal{G})) &= \log(\prod_{l=1}^{n}\prod_{i=1}^{p}\mathbb{P}(e_i^l|Pa(e_i^l))) \\
&= \log(\prod_{l=1}^{n}\prod_{i=1}^{p}\mathbb{P}(m_i^l, g_i^l|Pa(e_i^l))) \\
&= \log(\prod_{l=1}^{n}\prod_{i=1}^{p}\mathbb{P}(g_i^l|Pa(e_i^l), m_i^l)\mathbb{P}(m_i^l|Pa(e_i^l))) \\
&= \sum_{l=1}^{n}\sum_{i=1}^{p}\log(\mathbb{P}(g_i^l|Pa(e_i^l), m_i^l)) + \sum_{l=1}^{n}\sum_{i=1}^{p}\log(\mathbb{P}(m_i^l|Pa(e_i^l)))
\end{aligned}$$

avec $e_i^l = \{m_i^l, g_i^l\}$, l'observation du marqueur et du niveau d'expression du gène i pour l'individu l et $Pa(e_i^l)$, les observations des parents dans $\mathcal{G}$ du gène i pour l'individu l. La part de la liaison génétique entre marqueurs dans le calcul de la log-vraisemblance est contenu dans le terme $T = \sum_{l=1}^{n}\sum_{i=1}^{p}\log(\mathbb{P}(m_i^l|Pa(e_i^l)))$.

Afin d'ignorer cette dépendance induite par la liaison génétique, il convient de calculer la vraisemblance corrigée $\mathbb{P}'(\mathbf{D}|\mathcal{G}) = \prod_{l=1}^{n} \prod_{i=1}^{p} \mathbb{P}(g_i^l|Pa(e_i^l), m_i^l)$. Les différentes fonctions de score vues à la section 2.1.2.1 peuvent ainsi être adaptées à cette modélisation fusionnée en re-définissant $r'_i = r_{g_i}$ comme étant le domaine de définition de la seule variable d'expression g_i, $q'_i = r_{m_i} * \prod_{e_j \in Pa(e_i)} r_{e_j}$ le nombre de configurations possibles pour les parents de e_i en y incluant systématiquement le marqueur m_i et n'_{ijk}, le nombre d'occurrences de la configuration $(g_i = k, \{Pa(e_i), m_i\} = j)$. En d'autres termes la vraisemblance locale pour la variable e_i est calculée de manière similaire à la vraisemblance de g_i dans le modèle non fusionné où m_i est systématiquement inclus dans l'ensemble des parents. Paradoxalement le calcul du score local à e_i dans le modèle fusionné requiert ainsi une dissociation temporaire des variables m_i et g_i.

On peut alors voir le modèle fusionné comme un modèle non-fusionné contraint dans lequel chaque marqueur est parent de son gène associé et où l'ajout d'un nouveau parent pour g_i s'effectue uniquement par paire (m_j, g_j).

Le principal avantage de cette modélisation réside dans le nombre réduit de variables composant le graphe ce qui permet d'accélérer la recherche dans l'espace des structures possibles. Cependant cette fusion possède quelques inconvénients. Tout d'abord l'ajout d'une variable fusionnée en tant que parent équivaut à l'ajout simultané d'une variable d'expression et d'un marqueur, ce qui empêche de distinguer les différentes nuances de régulations (effet en *trans* ou *cis-trans*). De même la dimension du RB augmente plus fortement à chaque ajout ce qui, dans le cadre des scores pénalisés, limite l'apprentissage aux configurations locales ne présentant que très peu de parents. Par ailleurs la modélisation fusionnée ne permet pas de prendre en compte simplement les connaissances biologiques présentées pour le modèle non fusionné.

De manière générale le modèle fusionné représente une solution permettant d'intégrer les données de polymorphisme à moindre coût en terme de complexité mais qui, en contre partie, perd en terme de potentiel de description.

7.2 Comparaison des approches

Nous effectuons dans cette section différentes comparaisons portant à la fois sur les deux modélisations proposées et face à des méthodes classiques de reconstruction de RRG. Ces comparaisons ont été effectuées à partir de données simulées dont nous décrirons le processus de génération. Par la suite nous présenterons les résultats obtenus lors d'un des challenges de la compétition internationale DREAM5 en décrivant les différents méthodes proposées par notre équipe. Bien que les données simulées mises à disposition pour ce challenge soient similaires à celles déjà générées pour nos propres expérimentations, l'augmentation de la taille des réseaux proposés et le type de résultat attendu pour ce challenge nous ont amené à adapter notre approche.

Pour ces deux séries de comparaison, l'apprentissage s'effectue donc à partir de données simulées. L'utilisation de telles données permet de maîtriser les différents paramètres de simulation et de mesurer la robustesse des méthodes face aux variations de celles-ci. De plus la connaissance du vrai réseau simplifie l'évaluation des méthodes en apportant des critères objectifs de comparaison. Nous

proposerons une application sur des données réelles dans le chapitre 8.

Aparté sur l'apprentissage des réseaux bayésiens par BANJO Ce mémoire ne présente malheureusement pas les travaux dans l'ordre chronologique de leur réalisation. Le développement des nouveaux opérateurs présentés en partie 1 de même que l'utilisation du langage de programmation COMET ne sont intervenus que tardivement durant la thèse. Les comparatifs présentés ici étant antérieurs à ces travaux (hormis ceux de la section 7.2.2.6), certains choix techniques diffèrent donc des comparatifs présentés en première partie de ce manuscrit. Le principal changement réside dans l'implémentation des algorithmes d'apprentissage des RB. Ainsi nous utilisons dans cette section le logiciel BANJO proposé par Hartemink [2005] et fréquemment présenté comme logiciel de référence pour l'apprentissage de RRG dans le cadre des RB lors d'études comparatives. Le logiciel codé en JAVA, offre une vision compartimentée des méthodes de recherche locale pour l'apprentissage de structure, celle-ci s'articulant autour de 3 éléments à savoir : la définition du voisinage (*Proposer*), la politique de déplacement dans l'espace de recherche (*Decider*) et la fonction de score (*Scorer*). Chacun de ces 3 éléments peut être redéfini de manière indépendante permettant ainsi de mettre en place rapidement de nouvelles heuristiques de recherche ou d'implémenter de nouveaux scores.

Nous avons ainsi profité de ces facilités afin d'implémenter les scores BIC et $fNML$ dans ce logiciel venant s'ajouter au score *BDeu* déjà proposé. Nous avons également adapté ces scores afin de prendre en compte, le principe de score étendu, ainsi que le calcul dissocié pour le modèle fusionné.

Nous noterons bien évidemment que ce changement d'implémentation ne modifie aucun des résultats obtenus, hormis pour des considérations de temps d'exécution. L'implémentation telle que proposée dans le logiciel BANJO souffre malgré tout d'un défaut rédhibitoire, à savoir le re-calcul systématique de l'ensemble du voisinage après chaque modification locale. La redondance de ces calculs augmente ainsi le temps d'exécution de la recherche ce qui nous a amené à considérer d'autres pistes et à choisir de ré-implémenter totalement la méthode de recherche en langage COMET. Nous remarquerons cependant que cette approche permet de se libérer de la contrainte des mises-à-jour ciblées telles que nous l'avons justement proposé en COMET, offrant ainsi une implémentation plus générique.

Aparté sur la régression Lasso dans le cadre des données de génétique-génomique La majorité des approches comparées ici ne sont pas spécifiquement adaptées aux données de *génétique-génomique* ceci étant dû au nombre restreint de méthodes disponibles utilisant ce type de données. Ces méthodes utilisent donc les variables marqueurs comme des variables d'expression classiques et n'interdisent donc aucun type de relation à l'inverse de ce qui est pratiqué pour le modèle non-fusionné. S'appuyant sur le travail de plusieurs stages réalisés dans l'équipe SaAB portant sur la régression Lasso pour des données de *génétique-génomique*, nous avons souhaité étendre ces travaux à des fins de comparaison. Dans le cadre des systèmes d'équations linéaires, les marqueurs sont utilisés uniquement en tant que variables explicatives pour les variables d'expression. Chaque gène est donc régressé suivant les $2p - 1$ autres variables du modèle sans chercher à régresser les marqueurs. Cette modélisation, utilisée lors des différents stages, correspond aux premières restrictions biologiques du modèle non-fusionné.

La régression linéaire pour le gène G_i s'écrit alors

$$G_i = G\beta_i + M\alpha_i + \epsilon_i$$

où G_i est le vecteur des n observations du gène G_i, G et M sont des matrices $n \times p$ représentant respectivement l'ensemble des gènes et des marqueurs servant à prédire G_i tandis que β_i et α_i représentent respectivement les paramètres modélisant l'effet des variables d'expression et des marqueurs sur G_i avec une erreur gaussienne ϵ_i. Les paramètres β_{ii} sont fixés à 0 afin d'interdire la régression de G_i sur lui même. Nous noterons par ailleurs que pour tout paramètre estimé $\hat{\beta}_{ij} \neq 0$ correspond une arête $G_j - G_i$ dans le réseau et chaque paramètre estimé $\hat{\alpha}_{ij} \neq 0$ implique l'existence d'un arc $M_j \rightarrow G_i$.

Nous avons étendu ces premiers travaux afin d'appliquer les mêmes restrictions, issues de la position des marqueurs, que pour le modèle non-fusionné. Afin d'imposer la relation $M_i \rightarrow G_i$ il convient de régresser dans un premier temps G_i par rapport à M_i et de poursuivre la méthode standard en cherchant à estimer le résidu de G_i. A l'inverse interdire une relation $M_i \rightarrow G_j$ s'effectue simplement en retirant M_i de la liste des régresseurs possibles de G_j.

Dans les prochaines comparaisons portant sur les réseaux *Web-50* seuls le modèle non-fusionné et l'approche par régression Lasso bénéficieront donc de l'information portant sur la position des marqueurs. Nous testerons également la méthode SCT présentée en section 6.3 qui utilise des données de *génétique-génomique* pour la construction d'arbres de régression servant d'*a priori* pour une recherche de structure par MCMC. Bien que cette méthode distingue de fait les deux types de données, son implémentation ne permet pas la prise en compte des restrictions dues à la position des marqueurs.

7.2.1 Comparaisons sur les réseaux Web-50

7.2.1.1 Réseaux de régulations utilisés

Dans cette première série de comparaisons nous utilisons la collection Web50 proposée par Mendes et al. [2003] (accessible sur le site http://www.comp-sys-bio.org/AGN/), celle-ci est constituée de 50 réseaux de régulation artificiels. Chacun de ces réseaux contient p =50 gènes reliés par 50 arcs afin de former une structure dite *scale-free*, dont la distribution des connectivités des nœuds du graphe suit une loi de puissance. Cette structure caractéristique des réseaux biologiques réels connus actuellement [Barabasi and Oltvai, 2004] permet d'augmenter le réalisme des données simulées. Le réseau s'articule ainsi autour de quelques gènes *hubs* ayant un degré de connectivité élevé tandis qu'une grande majorité de leurs gènes satellites ne sont reliés qu'à un petit nombre de gènes. Afin de mieux rendre compte de la complexité des RRG, ces réseaux peuvent contenir à la fois des circuits ainsi que des régulations activatrices ou inhibitrices, le type de chaque régulation étant défini par tirage équiprobable. Un exemple de réseau de cette collection est présenté sur la Figure **7.5**.

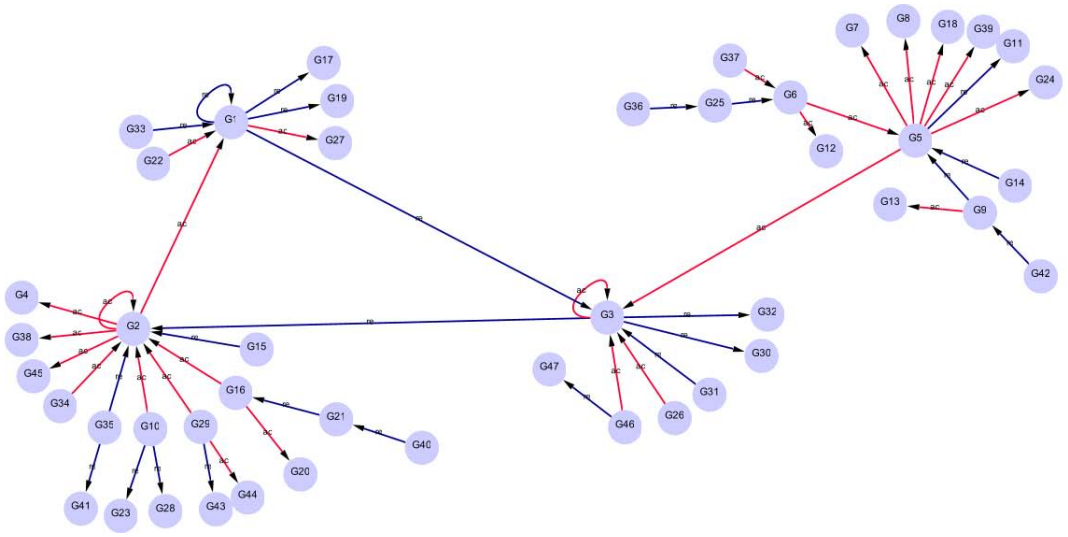


Figure 7.5 – Exemple du réseau de régulation de gènes artificiel Web50-001, les régulations rouges et bleues correspondent respectivement aux régulations activatrices et inhibitrices.

7.2.1.2 Génération des données de génétique-génomique

Définir la structure du réseau de régulation est une étape préliminaire à la génération des données, dans le cas des données de *génétique-génomique*, deux types de données sont nécessaires : les données de polymorphisme et les données d'expression. Chacune d'elles étant générée de manière spécifique.

Données de polymorphisme Les génotypes, c'est-à-dire l'état des SNPs (muté ou non), ont été obtenus grâce au logiciel CARTHAGENE [de Givry et al., 2005] qui permet, entre autre, de simuler les génotypes d'individus issus de divers types de croisement. Dans notre cas nous avons simulé pour chacun des 50 réseaux une population de $n = 500$ individus obtenus d'après un croisement de type *backcross*. Chaque individu de la génération courante est ainsi croisé avec l'un de ses parents. La population obtenue présente des génotypes variés pour chaque individu du au phénomène de *crossing-over*. Suivant notre hypothèse portant sur l'utilisation d'individus homozygotes, c'est-à-dire pour lesquels chaque chromosome d'une même paire présente les mêmes génotypes, seuls les génotypes d'un seul chromosome sont simulés pour chaque paire en supposant que le chromosome homologue présente ces mêmes génotypes. Nous supposons également l'existence d'un unique chromosome induisant ainsi un lien génétique continu sur l'ensemble des marqueurs.

Afin de respecter l'association 1 *gène* = 1 *marqueur*, 50 marqueurs SNP ont été répartis aléatoirement le long du chromosome de taille 10 Morgan. Chacun d'eux a ensuite été associé à l'un des 50 gènes suivant l'indice numérique de ces derniers dans la description du réseau. La position de chaque marqueur par rapport à son gène associé (région promotrice ou région codante) est tirée de manière équi-probable $(\frac{1}{2},\frac{1}{2})$. Nous noterons par ailleurs que cette position est commune à tous les individus du fait que chacun d'eux hérite des mêmes parents fondateurs. Dans ce premier comparatif,

nous supposons connues ces zones, toutefois nous verrons que cette information, inconnue dans le cas des données du challenge DREAM5, est retrouvée de manière efficace sans avoir recourt à une analyse eQTL. Finalement nous supposons que ces génotypes ne présentent aucune erreur ni valeur manquante.

Données d'expression Afin de simuler les données d'expression, nous avons suivi le protocole proposé par Liu et al. [2008] basé sur une fonction non linéaire de l'évolution du niveau de transcrit des gènes au cours du temps. Pour chaque gène est définie une équation différentielle ordinaire prenant en compte les activateurs et les inhibiteurs de ce gène, ainsi que la dégradation naturelle des molécules transcrites au cours du temps :

$$\frac{dG_i}{dt} = V_i \prod_{G_j \in Inh(G_i)} Z_j\left(\frac{1}{1+G_j}\right) \prod_{G_k \in Act(G_i)} Z_k\left(1+\frac{G_k}{G_k+1}\right) - G_i + \theta_i G_i$$

avec V_i, le taux moyen de transcription du gène i, $Inh(G_i)$ (resp. $Act(G_i)$), l'ensemble des gènes inhibiteurs (resp. activateurs) du gène i d'après le réseau artificiel choisi, Z_j (resp. Z_k), le taux moyen d'inhibition (resp. d'activation) de la transcription du gène i par le gène j (resp. k) et θ_i, un bruit gaussien de moyenne 0 et d'écart-type 0.1.

Les paramètres V et Z modélisent l'effet des mutations au niveau de l'expression des gènes. Pour chaque individu la valeur de ces paramètres varie en fonction de l'allèle observé aux marqueurs et de leur zone de localisation. Lorsque le marqueur M_i est situé dans la région promotrice de G_i alors le paramètre V_i est modifié en fonction de l'allèle observé. Plus exactement V_i vaut 1 si l'allèle de référence est observé sur M_i et β dans le cas de l'allèle mutant. Le marqueur M_i a donc un effet *cis* direct sur l'expression de G_i via V_i. A l'inverse si M_i est situé dans la région codante de G_i, le paramètre V_i est fixé à 1 pour tous les individus. Dans ce cas, c'est le paramètre Z_i qui est modifié en fonction de l'allèle observé. Nous remarquons que ce terme n'intervient pas dans l'expression de G_i. Le marqueur M_i n'a donc pas d'impact sur l'expression de G_i. Ce terme agit dans l'expression différentielle des gènes régulés par G_i, modélisant ainsi l'effet *trans*. Dans ces équations Z_i vaut 1 si l'allèle de référence est observé et β sinon.

Le paramètre β représente donc l'impact d'une mutation sur l'expression ou sur l'effet régulateur d'un gène, cet impact sera d'autant plus important que β diffère de 1. L'une des comparaisons présentées par la suite montrera l'effet de ce paramètre sur l'apprentissage.

Chaque gène possède ainsi une équation cinétique spécifique dont la valeur des paramètres varie pour chaque individu. Les états stationnaires de ces équations sont établis pour chacun des 500 individus à l'aide du simulateur de réactions biochimiques COPASI [Hoops et al., 2006]. Ces états stationnaires représentent les différents niveaux d'expression auxquels nous ajoutons une erreur expérimentale gaussienne centrée pour chaque gène G_i dont la variance est égale à 10% de la variance de l'expression de G_i.

A la différence des données de polymorphisme discrètes par nature, ce processus génère des

données d'expression continues. Afin d'appliquer certaines méthodes, dont celles basées sur les RB, nous discrétisons ces données en utilisant la méthode adaptative présentée en section 4.1.1.

7.2.1.3 Critères d'évaluation

A la différence des comparaisons présentées dans la partie 1, centrées sur l'apprentissage d'un RB, les méthodes comparées ici n'utilisent pas un même formalisme. Celles-ci peuvent donc être évaluées que sur le seul critère structurel. Le but est donc d'apprendre le graphe dont la structure est la plus proche du vrai réseau de régulation. Dans ce but, seules les régulations entre gènes nous importent, indépendamment du type de relation ayant permis de la déduire ($M \rightarrow G$ ou $G \rightarrow G$). Nous projetons donc pour toutes les méthodes (hormis le modèle fusionné) les relation de type $M_i \rightarrow G_j \forall j \neq i$ en tant que relation $G_i \rightarrow G_j$ et comparons le réseau appris uniquement sur ces relations entre gènes. De plus nous ne tiendrons pas compte de l'orientation des arcs du fait que la majorité des méthodes comparées ne fournissent pas de structures orientées.

Nous utilisons, dans ces comparaisons, les deux critères de précision et de sensibilité décrits dans la section 3.5.2 dont nous rappelons ici la description dans le cas des graphes non orientés.

- **Précision** $= \frac{VP}{VP+FP}$, indique le nombre d'arêtes correctes par rapport au nombre d'arêtes apprises ;
- **Sensibilité** $= \frac{VP}{VP+FN}$, indique le nombre d'arêtes correctes par rapport au nombre de vraies arêtes à apprendre ;

Les résultats présentés par la suite représentent la moyenne de ces deux critères sur les 50 réseaux appris à partir de chacun des jeux de données générés pour les 50 réseaux Web50.

7.2.1.4 Méthodes comparées

Les premières comparaisons s'effectuent uniquement dans le cadre des RB afin de comparer les deux modélisations puis d'analyser l'impact des scores étendus sur la reconstruction ainsi que l'apport des restrictions pour le modèle non fusionné. Nous utiliserons ces différents résultats afin de déterminer la configuration la plus performante dans le cadre des RB que nous comparerons à un ensemble représentatif de méthodes classiquement utilisées afin de reconstruire des RRG. Tout au long de ces comparatifs nous observerons également les performances des différents scores pour les RB et l'impact du nombre d'individus.

Afin d'apprendre la structure des deux modélisations proposées nous utilisons l'algorithme *Greedy Search* (GS) initialisé à partir du graphe vide. Comme précisé précédemment l'implémentation utilisée est celle disponible dans le logiciel BANJO. Dans le cas où le score *BDeu* est utilisé, nous fixons le paramètre $\alpha = 1$.

7.2.1.5 Résultats

Comparaison des deux modèles La Figure **7.6** montre une différence notable de qualité entre les réseaux appris grâce aux deux modélisations. Le modèle non fusionné obtient ainsi de meilleurs

résultats pour différentes tailles de population sur les deux métriques hormis pour le critère $fNML$ avec peu d'individus où le modèle fusionné garde une précision légèrement supérieure mais une sensibilité bien inférieure. Pour 50 et 100 individus, les réseaux sélectionnés par le modèle fusionné avec le critère BIC ne comportent que très peu d'arcs (2 et 5 en moyenne) ce qui explique la valeur de la sensibilité proche de 0. Ce résultat montre clairement la forte pénalité associée à la complexité du réseau appris avec ce type de critère. Cette pénalité est d'autant plus forte dans le modèle fusionné, en effet dans ce dernier la variable E_i représente l'ensemble des deux variables M_i et G_i, ainsi là où le modèle fusionné sélectionne E_i en tant que régulateur d'un gène, le modèle non fusionné autorise à ne retenir qu'une seule des deux variables si celle-ci se révèle suffisamment explicative. Ce dernier modèle permet ainsi de sélectionner un nombre plus important de régulateurs sans pour autant augmenter la dimension du réseau bayésien et donc d'améliorer la qualité des réseaux appris.

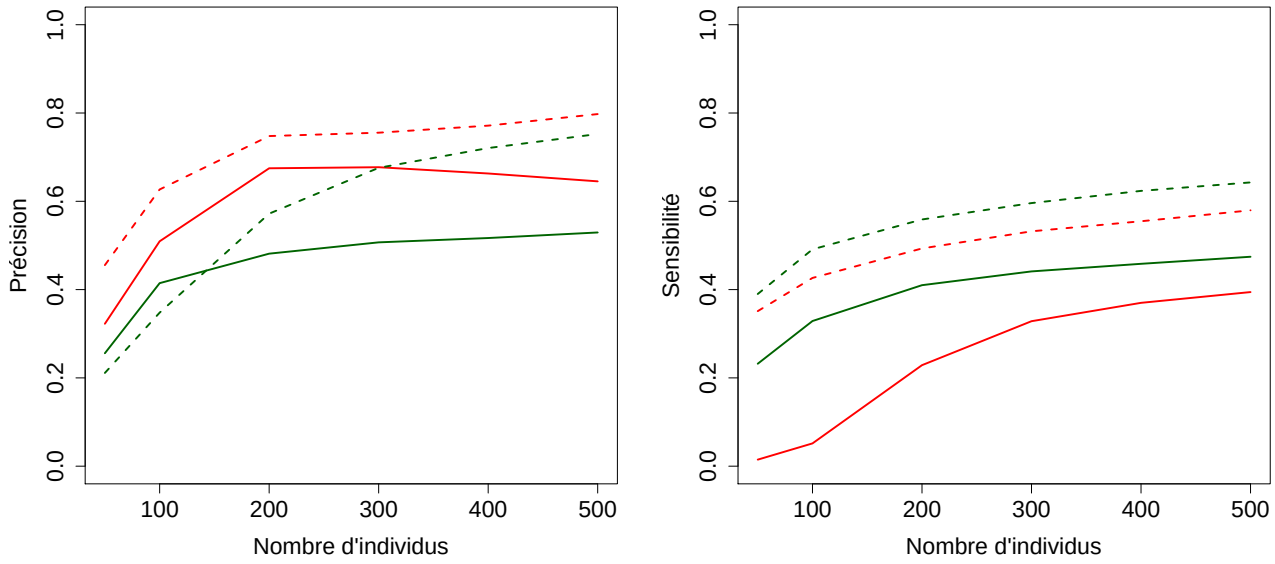


Figure 7.6 – Comparaison des modèles. Évolution de la précision (à gauche) et de la sensibilité (à droite) des réseaux appris pour les critères BIC_0 (en rouge) et $fNML_0$ (en vert), en fonction du nombre d'individus. Les modèles fusionné (ligne continue) et non fusionné (ligne traitillée) sont représentés pour chaque critère.

En nous basant sur ces premières observations, nous utiliserons par la suite uniquement le modèle non fusionné.

Impact des scores étendus Nous pouvons voir sur la Figure **7.7** un comportement similaire des scores BIC et $fNML$ (par souci de clarté nous ne présentons pas les résultats du score $BDeu$ similaire au BIC, hormis pour 50 individus où $BDeu$ se révèle moins performant). Dans le cas d'un petit nombre d'individus la hausse du paramètre γ permet d'améliorer de façon significative la précision des réseaux appris tandis que leur sensibilité se trouve en léger retrait. Ce comportement s'explique du fait qu'une

valeur élevée de γ favorise les structures à faible connectivité. Ce phénomène est d'autant plus marqué lorsque le nombre d'individus décroit, le graphe final comporte alors de nombreuses relations peu sûres dont une grande partie représente de fausses relations, augmenter γ permet de ne pas retenir ces arcs peu significatifs améliorant ainsi la précision du réseau final. La perte en sensibilité montre cependant que de vraies régulations figurent parmi ces relations supprimées. Malgré cela l'étude du ratio précision/sensibilité montre l'intérêt d'utiliser une valeur de gamma supérieure à 0. La valeur de ce ratio tend par ailleurs à se réduire lorsque γ se rapproche de 1, impliquant l'existence d'une valeur de γ optimale. Par ailleurs nous pouvons pressentir l'équivalence asymptotique des deux critères au vu de la convergence des résultats pour un nombre croissant d'individus tel que cela a été prouvé par Silander et al. [2010].

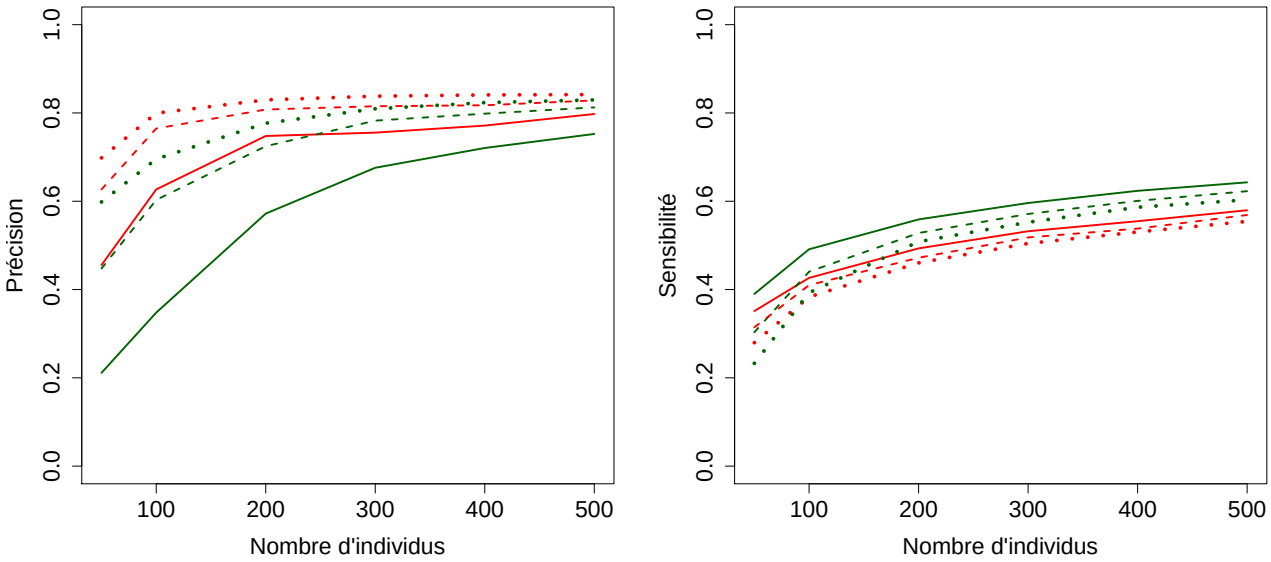


Figure 7.7 – Comparaison des critères étendus. Evolution de la précision (à gauche) et de la sensibilité (à droite) des réseaux appris avec les critères étendus *BIC* (en rouge) et $fNML$ (en vert), en fonction du nombre d'individus. Pour chaqu'un d'eux, 3 valeurs de gamma sont représentés : $\gamma = 0$ (ligne continue), $\gamma = 0.5$ (ligne traitillée) et $\gamma = 1$ (ligne pointillée).

Apport des données de génétique-génomique Grâce à certaines connaissances biologiques nous pouvons restreindre le nombre de structures possibles dans le cas du modèle non-fusionné comme nous l'avons montré dans la section 7.1.1. Nous comparons également la méthode basée sur p régressions Lasso indépendantes que nous avons adapté afin de prendre en compte ces informations.

Nous présentons sur la Figure **7.8** l'impact de ces connaissances sur la qualité des réseaux appris. Bien que ces restrictions ne donnent aucune information explicite sur les relations entre gènes, celles-ci orientent favorablement la recherche et améliorent la précision des différentes approches. Cette progression est d'autant plus significative que le nombre d'individus décroit, seule la méthode des

régressions Lasso se montre moins sensible avec 50 individus dû à ses mauvaises performances dans le cas de petites populations. De façon similaire à la section précédente le critère *BIC* et la méthode Lasso affichent une baisse de leur sensibilité suite à ces restrictions, celle-ci reste toutefois limitée pour le critère *BIC*. Seul le critère $fNML$ réagit positivement avec une amélioration de la qualité des réseaux sur les deux métriques, faisant de celui-ci un critère de choix pour incorporer des connaissances expertes supplémentaires.

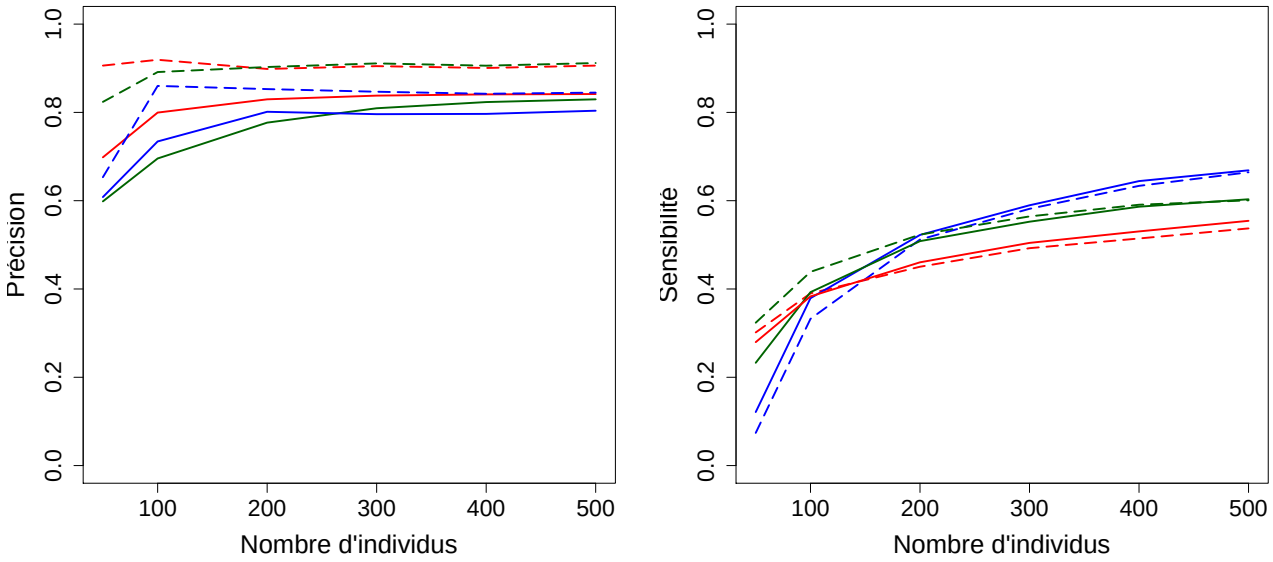


Figure 7.8 – Impact des connaissances biologiques. Évolution de la précision (à gauche) et de la sensibilité (à droite) des réseaux appris avec les critères BIC_1 (en rouge) et $fNML_1$ (en vert) ainsi que l'approche Lasso (en bleu), en fonction du nombre d'individus. Pour chaque approche, 2 tracés présentent l'apprentissage avec (ligne traitillée) et sans (ligne continue) ces restrictions.

Comparaison à l'état de l'art Nous comparons dans ce paragraphe le modèle non fusionné avec différentes approches présentées en section 6, nous étudions l'impact du nombre d'individus ainsi que la capacité des méthodes à gérer des données générées sous diverses conditions. Les logiciels ou packages utilisés sont disponibles gratuitement et résumés dans le Tableau **7.1**, accompagnés des choix effectués concernant divers paramètres dont les valeurs ont été définies de manière à obtenir un nombre d'arcs proche de celui du réseau réel. Nous utilisons pour le logiciel GGMSelect les familles "*C*01" et "*LA*", la famille "*EW*" ne donnant aucun résultat sur nos jeux de données. Compte tenu des précédents résultats, nous utilisons pour ce comparatif les critères BIC_1, $BDeu_1$, $fNML_1$ et l'approche Lasso en exploitant l'information de position des polymorphismes afin de restreindre les structures apprises possibles. Nous rappelons que dans notre comparatif seul notre modèle non fusionné, l'approche Lasso et le logiciel *SCT* sont aptes à distinguer les données d'expression et de polymorphisme, les autres méthodes ayant été développées dans le but de traiter uniquement les

niveaux d'expression, nous leur fournissons donc de manière indissociée les deux types de données. De plus seuls le modèle non fusionné et l'approche Lasso peuvent utiliser l'information portant sur la position des polymorphismes.

Logiciels	***Description***	***Paramètres***
BANJO(v2.2)	Réseaux bayésiens	$\alpha_{BDeu} = 1; \gamma = 1$
SCT(v0.1)	Réseaux bayésiens	$poids = \{1, 2\}; iterations = 10M$
ARACNE	Information mutuelle	$seuil = 0.15$
CLR(v1.2)	Information mutuelle	$seuil = 4$
ParCorA	Corrélation de Spearman	$pseuil = 0.01;\ 1er\ ordre$
SIMoNe(v1.0)	GGM	$nombredemodules = 2$
GGMselect(v0.1)	GGM	$familles\ C01\ \&\ LA$
GeneNet(v1.2.4)	GGM	$seuil = 0.95$
Rég Lasso	p régressions Lasso	$\alpha_{meinshausen} = 0.1$

Tableau 7.1 – Paramètres des méthodes testées

Nombre d'individus Le nombre d'individus disponibles pour l'apprentissage de structure reste bien souvent l'un des principaux facteurs limitants. Nous comparons ici le cas favorable d'une population de 500 individus à une situation plus réaliste où seuls 50 individus sont disponibles. En comparant les 2 colonnes de la Figure **7.9** nous pouvons noter de manière générale une grande disparité dans les performances des différentes approches, de même aucune d'entre elles ne semble clairement dominer ce comparatif. Deux groupes de méthodes émergent malgré tout de façon plus marquée lorsque le nombre d'individus augmente. La faible sensibilité des différentes approches avec une population réduite montre une incapacité commune à retrouver un nombre élevé de régulations. Ces difficultés proviennent notamment des structures de gènes en *hub* où un gène est fortement connecté aux autres gènes et dont l'expression se trouve dominée par un faible nombre de régulateurs occultant ainsi les autres régulations.

ARACNE et CLR obtiennent pour les différentes configurations des résultats similaires, CLR domine tout de même ce duel avec 500 individus face à ARACNE peu sensible à la taille de la population. Ces performances restent cependant éloignées de celle de ParCorA qui se montre efficace et semble tirer un avantage certain des mesures de corrélations partielles.

SIMoNe obtient les plus mauvais résultats du comparatif dû notamment à une faible précision, ce phénomène s'explique par le réseau modulaire autour duquel est conçu cette méthode, caractéristique ne correspondant pas à nos réseaux. En effet les nœuds des réseaux sont divisés en deux classes (niveaux d'expression et polymorphismes) mais en raison du nombre de régulations entre gènes dans nos données, les densités intra et inter classes sont très proches et ne permettent pas de distinguer d'éventuels modules (nous avons par ailleurs testés un nombre de modules variant de 2 à 6 sans aucune amélioration significative). La qualité des graphes résultants de GeneNet, GGMselect et Rég Lasso se dégrade rapidement lorsque la population diminue. Parmi ces 3 méthodes l'approche Lasso domine

pour 500 individus tandis que GGMselect tire partie d'une population plus petite, dans les différents tests ces deux approches gardent une précision élevée.

Notre modélisation apparaît comme une des approches les plus robustes de même que SCT et ParCorA, si ce dernier obtient d'ailleurs de meilleures performances pour 500 individus, la situation s'inverse pour 50 individus. Nous noterons au final qu'aucun des trois scores comparés ne se détache au niveau de la qualité des réseaux obtenus, bien que $BDeu_1$ domine légèrement les situations avec peu d'individus tandis que $fNML_1$ prend l'avantage lorsque la population croît.

Impact de la distance génétique entre marqueurs L'espacement des marqueurs le long du chromosome est un facteur influençant la difficulté d'apprentissage du réseau. Comme nous l'avons vu, la probabilité que l'état d'un marqueur (muté ou non) soit différent de celui qui le précède ou le succède sur le chromosome est directement proportionnelle à la distance génétique les séparant. Ainsi deux marqueurs espacés d'1cM auront des états différents sur seulement 1% de la population ce qui les rend potentiellement indissociables dans une population de 50 individus. Un espacement aléatoire produit notamment des distances entre marqueurs extrêmement faibles comparé à l'espacement uniforme qui représente le cas idéal. Ces faibles écarts tendent à perturber l'apprentissage du réseau en augmentant les ressemblances entre marqueurs, de fait, une erreur dans la sélection du marqueur responsable d'un effet *trans* peut conduire à une fausse régulation une fois la projection effectuée. Nous comparons les configurations (*a*) et (*b*) de la Figure **7.9** afin d'étudier cette perturbation.

On observe naturellement une diminution des performances pour toutes les méthodes lorsque l'espacement est aléatoire cependant cette baisse n'est pas identique suivant les approches. Alors que les performances des méthodes par RB et par corrélations ne se dégradent que légèrement, les approches basées sur un modèle linéaire montrent de fortes irrégularités. GeneNet perd sensiblement en précision avec peu d'individus tandis que SIMoNe se dégrade pour une grande population, seuls Rég Lasso et GGMselect restent faiblement perturbés.

Impact des mutations Lors de l'utilisation de données simulées, il est souvent difficile de quantifier le degré de réalisme de celles-ci, l'équation régissant l'expression d'un gène présentée précédemment n'échappe pas à cet aspect. Afin d'étudier la robustesse des approches à différentes données possibles sans remettre en cause la forme de l'équation différentielle, il convient de faire varier son unique paramètre β. Les configurations (b) et (c) de la Figure **7.9** présentent deux valeurs possibles de ce paramètre, nous rappelons qu'une valeur se rapprochant de 1 implique une réduction de l'effet des mutations sur l'expression des gènes.

Au niveau des différentes méthodes, on observe une stabilité générale des méthodes de corrélations ainsi que de SCT tandis que notre approche se dégrade de façon plus marquée tout en restant compétitive pour 50 individus. Les performances des méthodes Rég Lasso et GGMselect se dégradent quant à elles pour 500 individus (fortement pour ce dernier) et augmentent légèrement pour 50 individus. Seul SIMoNe se détériore pour les deux tailles d'échantillon. Les résultats de GeneNet sont les plus surprenants avec une précision divisée par deux pour 50 et 500 individus, tandis que cette baisse se trouve compensée, uniquement dans le cas de la plus faible population, par une hausse de la sensibilité gardant ainsi le même ratio précision/sensibilité.

Les dégradations observées lorsque $\beta = 0.25$ s'expliquent par une domination accrue des marqueurs sur les niveaux d'expression, masquant ainsi les autres régulations. Cette même difficulté se retrouve au niveau des *hubs*. Une valeur informative de β ne devra donc pas être trop proche de 1 au risque de rendre invisible l'effet des mutations, mais ne devra pas en être trop éloignée afin d'éviter une domination des marqueurs. Ce phénomène s'amplifie d'autant plus pour les gènes dont les marqueurs sont situés en région promotrice.

Malgré ces résultats hétérogènes, certains points ressortent de ces comparatifs. Les différents scores utilisés pour sélectionner les réseaux bayésiens avec l'algorithme GS obtiennent des résultats similaires et se montrent particulièrement stables dans les différentes configurations testées de même que SCT. Les méthodes de corrélations apparaissent également robustes, seules les méthodes Rég Lasso et GGMselect montrent un caractère semblable parmi les modèles linéaires tandis que les autres méthodes se révèlent particulièrement instables. Parmi toutes ces méthodes, ParCorA et les approches par réseaux bayésiens (GS et SCT) obtiennent au final les meilleurs résultats cumulés de cette comparaison sur données simulées. Cependant SCT nécessite le réglage de 8 paramètres différents dont la pondération de l'*a priori* à laquelle la méthode se révèle très sensible. Bien que nous utilisions ici une pondération adaptée connaissant les vrais réseaux, le réglage des différents paramètres lors de l'analyse de données réelles peut s'avérer difficile.

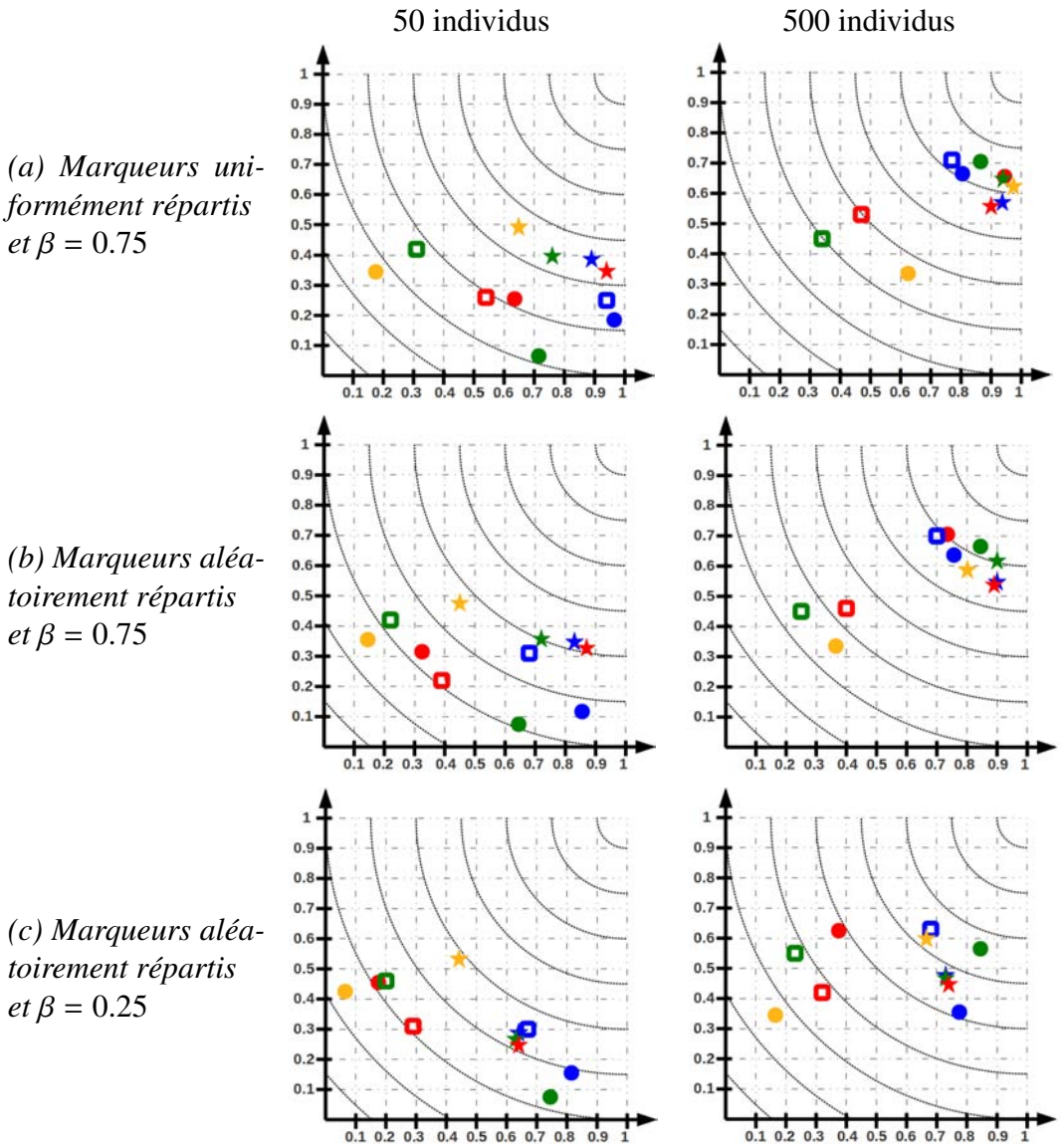


Figure 7.9 – Précision (axe horizontal) et sensibilité (axe vertical) pour 50 (colonne de gauche) et 500 individus (colonne de droite) dans 3 configurations. **(a)**Les marqueurs sont espacés uniformément sur le chromosome avec un paramètre β quantifiant l'impact de la mutation égal à 0.75. **(b)**Les marqueurs sont espacés aléatoirement avec le même β. **(c)**Les marqueurs sont espacés aléatoirement et les mutations ont un impact plus fort ($\beta = 0.25$). Pour chaque figure les méthodes sont classées en 3 catégories : **1-Réseaux de corrélations** ARACNE (carré vert), CLR (carré rouge), ParCorA (carré bleu) ; **2-Modèles linéaires** SIMoNe (cercle jaune), GeneNet (cercle rouge), GGMSelect (cercle bleu), Lasso (cercle vert) ; **3-Réseaux bayésiens** SCT(étoile jaune) et BANJO avec BIC_1 (étoile rouge), $BDeu_1$ (étoile bleue), et $fNML_1$ (étoile verte).

7.2.2 Compétition DREAM

Nous nous intéressons cette fois aux résultats obtenus par notre équipe lors d'un challenge de la compétition DREAM5, dont l'objectif était de reconstruire 15 RRG à partir de données de *génétique-génomique*. Nous avons proposé pour ce challenge, trois approches distinctes ainsi qu'une 4ème méthode dite de *méta-analyse* qui effectue le consensus des résultats des trois premières approches. Cette *méta-analyse* a permis d'obtenir les meilleurs résultats de ce challenge. Nous présenterons donc successivement les objectifs de cette compétition et les données utilisées pour le challenge puis nous décrirons les méthodes proposées ainsi que les résultats obtenus.

Les méthodes et résultats présentés ici sont issus du travail collectif de l'équipe SaAB et ont été publiés dans Vignes et al. [2011]. Chacune de ces méthodes a été développée par différents membres de l'équipe, ma contribution personnelle ayant naturellement porté sur l'approche basée sur les RB.

7.2.2.1 Présentation de la compétition DREAM

Depuis 2006, la compétition internationale DREAM [Stolovitzky et al., 2007] propose chaque année différents challenges visant à une meilleure compréhension des réseaux biologiques. Ces challenges permettent d'effectuer un état des lieux et de comparer les méthodes existantes autour de l'apprentissage des réseaux d'interactions et de la prédiction de leur dynamique. La variété des sujets abordés permet par ailleurs de couvrir un large spectre des problématiques impliquant les différents acteurs de la vie cellulaire (protéine, ARN, séquence d'ADN) au travers de données simulées ou réelles. En 2010, l'édition DREAM5 proposait un challenge intitulé *Systems Genetics* donc l'un des objectifs fût justement de reconstruire des RRG à partir de données de *génétique-génomique*.

7.2.2.2 Données du challenge

Réseaux à apprendre L'objectif du challenge *SysGenA* était d'apprendre la structure de 15 réseaux composés chacun de p =1 000 gènes et dont le nombre d'arcs varie environ entre 2 000 et 5 000. La distribution des connectivités diffère selon le caractère entrant ou sortant des degrés, ainsi les distributions de ces degrés suivent respectivement des lois de puissance et des lois exponentielles, le tout devant former une structure modulaire.

Trois sous-challenges étaient définis suivant le nombre d'échantillons disponibles pour l'apprentissage. Ainsi les trois sous-challenges A100, A300 et A999 sont composés pour chacun de 5 réseaux dont la structure doit être apprise à partir d'un échantillon de taille respective n =100, 300 et 999. Le nombre d'arcs des 5 réseaux d'un même sous-challenge croit progressivement, permettant ainsi d'analyser les performances de reconstruction pour différentes configurations de la taille de l'échantillon et de la connectivité du réseau. On notera également que le rapport entre le nombre d'échantillons et le nombre de gènes reste strictement inférieur à 1 ce qui nous place dans un problème de grande dimension, typique des situations réelles.

Simulation des données La procédure utilisée afin de simuler les données de *génétique-génomique* présente de fortes similitudes avec celle présentée dans le cas des réseaux *Web50*. Pour chaque réseau, une population de n individus issue d'un croisement de type *RIL* a été simulée puis génotypée à

l'aide de 1000 marqueurs bi-alléliques situés aléatoirement sur 20 chromosomes distincts. Chacun de ces marqueurs est associé à un gène et se situe dans la région promotrice ou codante de celui-ci. A la différence de nos propres simulations, la localisation des marqueurs n'était pas connue à l'avance, seules les proportions de marqueurs situés dans la région promotrice et codante étaient fournies (respectivement 25% et 75%). Cette répartition non-uniforme des deux positions s'explique par une longueur des régions promotrices généralement inférieure à celle des régions codantes, ainsi la probabilité qu'un marqueur soit situé dans une région codante est logiquement plus élevée.

Les données d'expression sont simulées à l'aide d'une équation différentielle ordinaire dont l'expression est semblable à celle présentée précédemment et qui inclue les mêmes paramètres modélisant l'effet des mutations. L'équation utilisée comporte cependant certains paramètres supplémentaires comme un coefficient de *coopération* ou une *constante de Michaelis* dont les valeurs sont tirées aléatoirement suivant des distributions de probabilité standards (uniforme, gaussienne ou gamma). Ces paramètres ont pour objectif principal de faire varier la puissance de chaque régulation afin de simuler plus finement la dynamique d'un vrai RRG.

L'intégralité de ce processus regroupant à la fois la génération des réseaux et des données de *génétique-génomique* est disponible au sein du logiciel SysGenSIM [Pinna et al., 2011].

Comparaison avec les réseaux Web50 Les données simulées dans le cadre de ce challenge présentent de fortes similarités avec celles générées pour les réseaux *Web50* du fait de la même association *1 gène-1 marqueur* ou bien de l'équation différentielle simulant la dynamique des expressions. La principale différence réside dans les dimensions des réseaux proposés et de leur densité. Le nombre de gènes est ainsi 20 fois supérieur au cas des réseaux *Web50*, de même que la connectivité, en hausse d'un facteur allant de 2 à 5. Le nombre d'échantillons est également plus réaliste ce qui permet de se confronter aux problèmes inhérents de la grande dimension.

D'autres différences spécifiques aux données de polymorphisme peuvent également être relevées, notamment la répartition des marqueurs sur 20 chromosomes distincts. Cependant la simulation sur plusieurs chromosomes n'a pas fondamentalement transformé la difficulté d'apprentissage, en créant 20 blocs indépendants de marqueurs (les chromosomes) elle a pu légèrement simplifier le choix des marqueurs en *trans* en réduisant les confusions possibles par rapport à un seul chromosome. De même la zone de localisation des marqueurs bien qu'inconnue dans ce cas, peut être retrouvée à l'aide d'une analyse eQTL, d'autant plus lorsque l'effectif de chacune des deux zones est connu à l'avance. Cependant pour ces données, nous avons employé une méthode plus simple permettant de distinguer efficacement les deux types de polymorphisme, méthode que nous décrivons dans le paragraphe suivant.

Zone de localisation des marqueurs Nous avons effectué pour chaque gène une analyse de variance afin de détecter les corrélations significatives entre celui-ci et son marqueur associé. Un marqueur est alors défini comme *cis* dès lors que que cette corrélation est significative, tout en s'assurant qu'aucun des 3 marqueurs voisins en amont et en aval sur le chromosome n'est davantage corrélé à ce gène. Ce principe de fenêtre sur le chromosome, fréquemment employé en génétique, permet de limiter le biais induit par le lien génétique entre les marqueurs proches sur un même

chromosome. La *p-value* du test de corrélation nous a permis de classer les marqueurs en 3 classes à l'aide de deux seuils α_1 et α_2 ($\alpha_1 \leq \alpha_2$) tels que : *p-value* $< \alpha_1$ indique que le marqueur est en *cis*, *p-value* $> \alpha_2$ indique un marqueur en *trans* et lorsque $\alpha_1 \leq$ *p-value* $\leq \alpha_2$ le marqueur est de type *indéterminé*.

A l'aide de ce test nous obtenons pour $\alpha_1 = 0.001$ et $\alpha_2 = 0.9$, une répartition des 1000 marqueurs suivant leur type (*cis*/*trans*/*indéterminé*) moyennée sur les 5 réseaux pour chaque sous-challenge :

- A999 (240/618/142),

- A300 (219/653/128),

- A100 (127/672/201).

Ce test permet de retrouver, avec un nombre d'individus suffisant, une proportion de *cis* proche des 25% attendus (24% et 22% pour 999 et 300 individus) tandis que cette proportion s'éloigne clairement pour 100 individus. Nous utilisons la classification issue du test pour appliquer les restrictions vues précédemment aux marqueurs de type *cis* et *trans*. Les marqueurs de type *indéterminé* ne sont quant à eux soumis à aucune restriction sur leurs arcs sortants.

7.2.2.3 Critères d'évaluation

Pour le challenge chaque méthode devait fournir pour chaque réseau une liste des 100 000 premières régulations orientées entre gènes et classées suivant un indice de confiance. Afin d'évaluer la qualité d'une méthode à partir de cette liste, plusieurs techniques peuvent être employées. La première d'entre-elles consiste à fixer un seuil unique afin de construire un graphe final composé des arcs dont l'indice de confiance est supérieur à ce seuil. Chaque méthode est alors comparée sur la base de ce graphe. Cependant l'utilisation de méthodes variées rend difficile l'uniformisation des indices de confiance et donc peu judicieuse l'idée d'une comparaison sur un graphe unique. Plutôt que de considérer l'indice de confiance des régulations, seul leur classement suivant cet indice peut être utilisé. A partir de ce classement le seuil représente alors un rang au delà duquel un arc est retenu dans le graphe final. En faisant varier ce seuil on obtient un ensemble de graphes de connectivité progressive, la qualité structurelle en terme de précision et de sensibilité de chacun de ces réseaux, permet de tracer une courbe comparable entre les différentes méthodes. Cette courbe évite ainsi l'écueil lié au choix du seuil de confiance ou d'un paramètre de l'algorithme gérant la stringence de la méthode.

Deux courbes ont été utilisées afin de comparer les méthodes :

- la courbe ROC (*Receiver Operating Curve*) représente l'évolution de la *sensibilité* en fonction du *taux de faux positif = (1-spécificité)*,

- la courbe PR (*Precision Recall*) représente l'évolution de la *précision* en fonction de la *sensibilité*.

Intuitivement la courbe ROC indique la proportion d'arcs du vrai réseau appris (avec raison) en fonction de la proportion des non-arcs du vrai réseau appris (à tord). Ainsi le tracé d'une méthode classant des vrais arcs dans les premières places va donc augmenter sa sensibilité rapidement sans pour autant augmenter son taux de faux positif.

A l'inverse la courbe PR indique la proportion d'arcs appris avec raison en fonction de la proportion d'arcs du vrai réseau appris (avec raison), c'est-à-dire que chaque ajout d'un arc au réseau suivant le classement, permet d'en augmenter la sensibilité sans pour autant dégrader la précision.

Une fois ces courbes tracées, la mesure de l'aire sous chacune d'elles permet de quantifier la performance d'une méthode. Une méthode est ainsi d'autant plus performante que ces deux aires sont grandes. Les organisateurs du challenge ont par ailleurs calculé des *p-value* à partir de ces aires afin de mesurer les performances de chaque méthode relativement aux autres approches. Ces *p-value* indiquent ainsi la probabilité que les vrais arcs appris par une méthode pour un rang donné, puissent être obtenus par tirage aléatoire dans l'ensemble des arcs de même rang appris par les autres méthodes. En d'autres termes ces *p-values* caractérisent la capacité d'une méthode à retrouver des régulations qu'aucune autre approche n'a appris à un rang donné.

7.2.2.4 Méthodes comparées

Pour ce challenge l'équipe SaAB a proposé 4 méthodes, dont l'une vise à effectuer la synthèse des 3 autres. Nous présentons dans cette section ces différentes méthodes à savoir une approche basée sur les RB, deux méthodes modélisant le problème sous forme d'équations linéaires résolues à l'aide de régressions de type Lasso ou sélecteur de Dantzig ainsi que la méta-analyse des trois méthodes précédentes.

Réseaux bayésiens Nous avons utilisé le modèle non-fusionné afin d'apprendre le RB, au vu des meilleurs performances de celui-ci comparé au modèle fusionné. Les différentes restrictions liées aux réalités biologiques ainsi qu'à la position des marqueurs, sont appliquées de la même manière, hormis dans le cas des marqueurs indéfinis pour lesquels aucune restriction sur les arcs sortants n'est formulée. La discrétisation des données d'expression s'effectue toujours suivant la méthode adaptative.

Deux changements notables ont tout de même été appliqués par rapport au comparatif précédent. Le premier d'entre eux provient de l'augmentation de la dimension des réseaux à apprendre, nécessitant de fait l'exploration de voisinages plus larges. Cette augmentation est d'autant plus sensible avec le logiciel BANJO où l'intégralité de ce voisinage est recalculée après chaque modification du graphe. Pour cela nous avons restreint l'espace de recherche une seconde fois en appliquant le filtre présenté en section 4.2. Nous conservons ainsi comme parent potentiel pour chaque gène l'ensemble des variables d'expression ou de marqueur permettant d'améliorer le score comparé à la situation où ce gène n'a aucun parent. De plus afin d'éviter qu'une succession de marqueurs sur un même chromosome ne soit sélectionnée pour un même gène du fait des corrélations issues du lien génétique, nous utilisons un système de fenêtre glissante. Seul le meilleur marqueur à l'intérieur de cette fenêtre est retenu dans la liste des parents potentiels ce qui assure une distance minimum entre chaque

marqueur sélectionné. Nous avons fixé la taille de cette fenêtre à 50cM, afin que deux marqueurs successifs retenus puissent être considérés comme indépendants. Ce filtre intervient en complément des restrictions biologiques appliquées précédemment, celui-ci ne remet donc pas en cause les arcs imposés ou interdits dans le réseau.

Le second changement est issu de la nécessité de retourner une liste triée des 100 000 premières régulations entre gènes. Ce type de résultat s'obtient facilement à l'aide de méthodes d'apprentissage ensemblistes telles que l'algorithme MCMC ou toute autre méthode stochastique, en utilisant la fréquence d'apparition des arcs dans les graphes sélectionnés. Cependant le caractère déterministe de l'algorithme GS implémenté dans le logiciel BANJO, ne permet pas d'obtenir un classement aussi large. Afin de proposer une liste d'arcs ne se résumant pas à un unique réseau nous avons choisi d'apprendre plusieurs réseaux à l'aide du score *BDeu* en faisant varier le paramètre α. Ce paramètre qui modélise un *a priori* sur l'existence d'individus pour lesquels le graphe est complet, a pour effet d'augmenter la connectivité du RB appris lorsque sa valeur augmente [Silander et al., 2007]. Nous avons donc fait varier α suivant une grille de 20 valeurs dans l'ensemble des puissances de 10 $\{10^{-16}, 10^{-15}, ..., 10^{3}\}$. Après projection des relations issues des marqueurs pour les 20 réseaux, les régulations entre gènes ont été classées suivant leur nombre d'occurrences.

Lorsque plusieurs régulations apparaissent un même nombre de fois, celles-ci sont départagées d'après leur score d'influence calculé par le logiciel BANJO. Ce score proposé par Yu et al. [2002] quantifie la force de chaque arc du graphe. Pour cela seules les classes extrêmes des discrétisations sont considérées afin d'obtenir une discrétisation binaire de type (*Bas*/*Haut*), les états intermédiaires étant ignorés. Le score d'influence d'une régulation est alors défini suivant le nombre d'observations où les deux variables reliées sont toutes deux dans le même état ou dans l'état inverse. Un score d'influence élevé indique par exemple que les deux variables ont été observées simultanément dans leur état *Haut* un grand nombre de fois. A la différence du score *BDeu* ce score ne caractérise que des relations linéaires ce qui justifie son rôle d'arbitre dans le cas de relations équi-fréquentes uniquement.

Bien que cette procédure permet de classer un plus grand nombre de régulations, celui-ci reste encore insuffisant. Ainsi le classement le plus large des 15 réseaux appris ne comporte que 5 000 relations, bien en deçà donc des 100 000 régulations attendues. Nous verrons lors des résultats la conséquence de ce classement partiel. Par ailleurs l'utilisation du paramètre γ afin de restaurer un même *a priori* sur les classes de connectivité, ne se révèle pas pertinente dans cette situation, du fait qu'augmenter sa valeur amène à apprendre des structures moins denses ce qui va à l'encontre d'une volonté de classer d'un grand nombre de régulations. Une autre solution consiste à utiliser des valeurs élevées du paramètre α, cependant cette valeur ne peut être augmentée de manière inconsidérée au risque d'apprendre un réseau dénué de sens. Ainsi pour notre valeur de α maximum ($\alpha = 10^3$), le nombre d'individus *a priori* est déjà supérieur au nombre d'individus réels de chaque sous-challenge, augmenter davantage sa valeur conduirait à apprendre des réseaux fortement connectés uniquement dus à ces individus virtuels.

Régression Lasso Nous avons déjà abordé la modélisation du problème d'apprentissage des RRG dans le cadre de régressions indépendantes de type Lasso (voir section 6.2.2), notamment pour des données de *génétique-génomique*. Dans ce cas chaque gène est régressé sur l'ensemble des $2p - 1$

autres variables du modèle. A la différence de la comparaison portant sur les réseaux *Web-50*, les régressions Lasso n'ont pas été restreintes à l'aide des positions des polymorphismes. Du point de vue de la mise en œuvre, ces régressions ont été effectuées à l'aide de l'algorithme *LARS* inclus dans le package *glmnet* sous R.

Afin de classer le maximum d'arcs, 20 réseaux ont été reconstruits à partir de différentes valeurs de la pénalité λ de manière similaire à l'approche suivie pour les RB. Les 20 valeurs de λ sont réparties uniformément dans l'intervalle $[0,\lambda_{max}]$. Lorsque $\lambda = 0$ aucune pénalité n'est appliquée, toutes les variables rentrent donc dans le modèle, ce qui équivaut au graphe complet, à l'inverse λ_{max} représente la valeur à partir de laquelle plus aucune variable explicative n'entre dans le modèle.

L'une des difficultés des modèles linéaires pour ce challenge réside dans le fait que les relations apprises ne permettent pas de déterminer le sens des régulations entre gènes, seules les relations issues d'un marqueur peuvent être orientées. Afin d'effectuer un classement des régulations orientées au plus juste il a été choisi de pondérer dans le réseau final les relations apprises suivant leur type. Dans un premier temps chaque relation orientée provenant d'un marqueur de type $M_i \rightarrow G_j$ est projetée avec une pondération de 1 en tant que régulation $G_i \rightarrow G_j$. Puis chaque relation non-orientée apprise de type $G_i - G_j$ pour laquelle aucune des deux relations $M_i \rightarrow G_j$ ou $M_j \rightarrow G_i$ n'a été apprise, est projetée avec une pondération de 0.5 pour les deux orientations $G_i \rightarrow G_j$ et $G_j \rightarrow G_i$. La moyenne de ces 20 graphes orientés pondérés permet alors de fournir le classement attendu. Ce même système de pondération est appliqué au sélecteur de Dantzig présenté immédiatement après.

Sélecteur de Dantzig Le sélecteur de Dantzig offre une alternative intéressante afin d'effectuer les p régressions linéaires. De la même manière que le Lasso, le sélecteur de Dantzig utilise une pénalité de type ℓ_1 sur les paramètres mais présente le problème sous un autre angle. On cherche ainsi à estimer les paramètres tels que

$$\hat{\beta}_i(t) = \text{argmin}_{\beta \in \mathbb{R}_p} \{ \| \beta_i \|_{\ell_1} \} \quad \text{sous contrainte } \| {}^\top X(G_i - X\beta_i)\|_{\ell_\infty} \leq \delta$$

où G_i est le vecteur des observations du gène G_i, X la matrice $n \times 2p$ de l'ensemble des observations et β_i le vecteur des coefficients d'interaction de G_i avec les p gènes et p marqueurs du réseau.

Cet estimateur cherche donc à minimiser la pénalité sur les paramètres sous contrainte d'une borne supérieur sur la corrélation entre les variables explicatives et les résidus. On cherche donc à sélectionner le modèle possédant le moins de variables explicatives tout en assurant que la part des résidus restant à estimer, ne dépasse par un seuil fixé. Pour $\delta = 0$ la contrainte implique que les résidus de la régression ne peuvent pas être mieux estimés ce qui correspond à la présence de toutes les variables dans le modèle. Lorsque $\delta \rightarrow \infty$ la contrainte sur les résidus se relâche totalement, les coefficients β_i sont alors mis à zéros.

L'estimation des paramètres de chacune de ces p régressions peut se réduire à un problème d'optimisation linéaire se résolvant à l'aide d'algorithmes dédiés. Pour le challenge, le programme *glpk* a été choisi dans ce but. De la même manière que pour les deux méthodes précédentes, ces

régressions ont été effectuées pour 20 valeurs équidistantes sur l'intervalle [0,δ_{max}] où δ_{max} représente la première valeur pour laquelle plus aucune variable n'entre dans le modèle.

Méta-analyse La dernière approche proposée consiste à effectuer le consensus des 3 méthodes présentées précédemment. Pour cela nous avons utilisé un meta-test de Fisher dont le but originel est de combiner des *p-values* issues de tests indépendants. Dans notre cas la pondération de chaque régulation peut être assimilée à une probabilité d'apparition d'un arc correspondant ainsi à (1−*p-value*).

Nous calculons alors la valeur du meta-test pour chaque régulation de G_i vers G_j

$$S_{ij} = \sum_{m \in \mathcal{M}} \log(1 - r_{ij}^m)$$

où $\mathcal{M} = \{BN, Lasso, Dantzig\}$ représente l'ensemble des 3 méthodes à combiner et r_{ij}^m la pondération pour la méthode m sur la régulation $G_i \rightarrow G_j$.

Afin de retrouver une pondération des régulations qui range les arcs du plus probable au moins probable, le pondération $r_{ij}^{\mathcal{M}}$ consensus est telle que

$$r_{ij}^{\mathcal{M}} = 1 - \exp(S_{ij})$$

7.2.2.5 Résultats officiels

Nous commençons par présenter les résultats officiels obtenus lors de cette compétition, puis nous analysons plus particulièrement le comportement de la méta-analyse et la cohérence des trois approches sur lesquelles celle-ci repose.

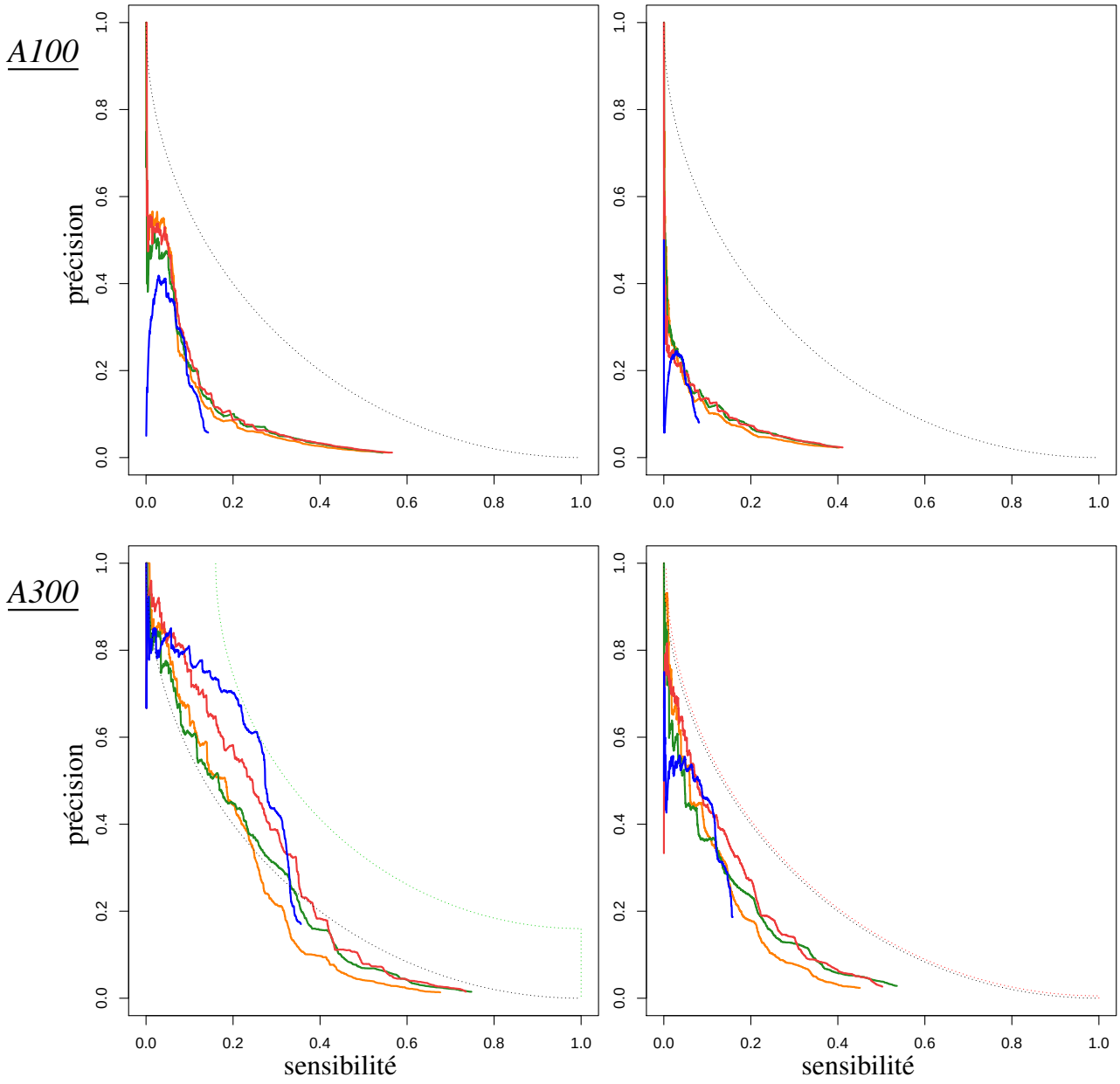
Courbes précision-sensibilité Les courbes précision-sensibilité présentées sur la Figure **7.10** permettent de visualiser rapidement l'impact du nombre d'individus et de la densité du réseau sur les performances de chacune des méthodes. En toute logique les meilleurs résultats sont obtenus pour le réseau le moins dense avec 999 individus (premier réseau A999). Dans cette situation, un premier classement se dégage visuellement, dominé par la méta-analyse, suivit par le sélecteur de Dantzig qui dépasse les RB au début de la courbe avant que la situation ne s'inverse à partir d'une sensibilité de 0.2. La régression Lasso ferme la marche, malgré une baisse de la précision moins rapide que les deux autres approches lorsque la sensibilité est supérieure à 0.5. Ce classement visuel dépend fortement des attentes de l'utilisateur au niveau du réseau prédit. Dans notre cas nous nous basons sur la distance minimum atteinte par rapport au point correspondant à une précision et une sensibilité de 1 (situation du vrai réseau). D'autres attentes mènent à préférer la méthode permettant de retrouver le maximum de vraies régulations en maintenant une précision au dessus d'un seuil fixé, ou à l'inverse de souhaiter la précision la plus haute possible pour une sensibilité donnée. Lors d'applications réelles, dû au coût élevé pour chaque validation expérimentale, les méthodes garantissant une bonne précision sont

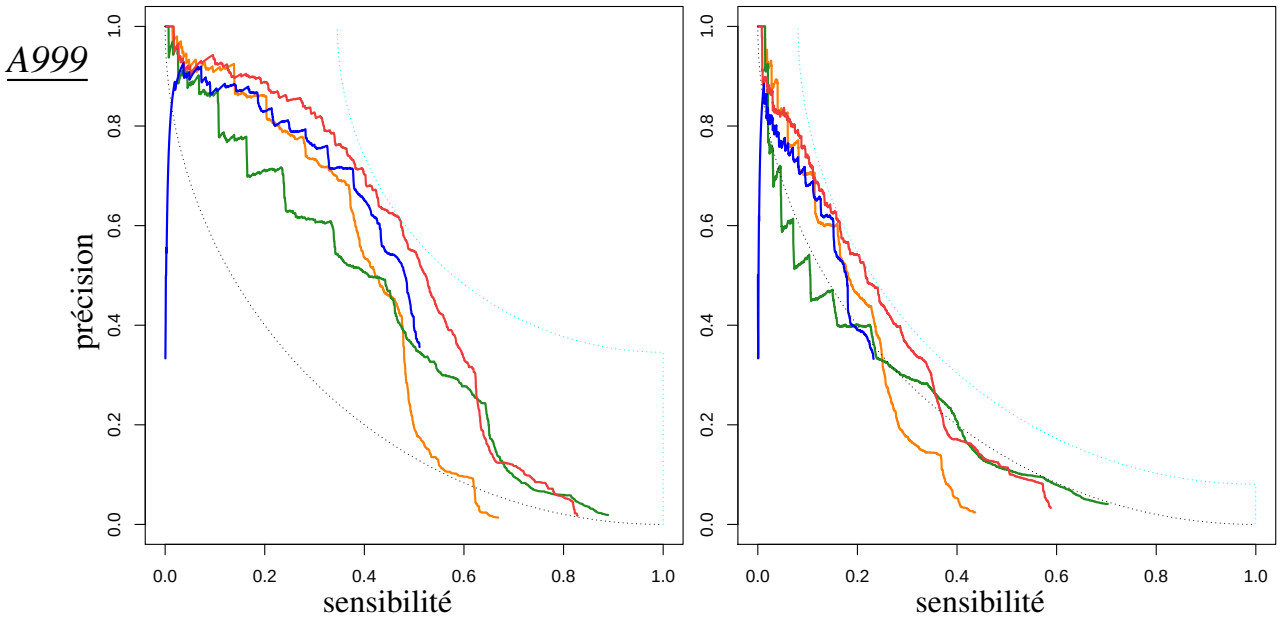
préférées en dépit de leur sensibilité plus limitée.

Dans le cas du challenge les méthodes ont été comparées suivant un critère plus neutre calculant les aires sous ces courbes. Cette mesure pénalise cependant les méthodes ne pouvant classer 100 000 régulations. Comme nous pouvons le constater sur la Figure **7.10**, les courbes des RB se terminent avant même d'avoir atteint une précision proche de 0 qui représente la qualité du graphe comportant 100 000 régulations. Dans cette situation, les organisateurs ont décidé de compléter les classements partiels en tirant aléatoirement parmi les régulations possibles. Un tel tirage revient alors à poursuivre le tracé de la courbe interrompue, par une ligne quasi-verticale (la probabilité de tirer une vraie régulation parmi l'ensemble des possibles étant très faible) jusqu'à atteindre une précision proche de 0. On imagine alors aisément la perte, en terme d'aire sous la courbe, due à cette complétion.

Se détachant de ces mesures d'aire qui de plus sont difficilement évaluables visuellement, nous comparons les approches suivant leur distance minimale au point (sensibilité=1,précision=1). Dans le cas des plus grandes populations (A999), un classement clair se précise comme mentionné précédemment. Cependant l'écart entre les méthodes se réduit lorsque la connectivité du vrai réseau augmente dû à une baisse notable de leur sensibilité. La régression Lasso atteint, pour les deux réseaux, la plus haute sensibilité. On remarque de manière générale le comportement singulier des RB au niveau du premier point de chaque tracé. Ce point reflète la qualité du réseau composé uniquement de la première régulation classée par la méthode. Alors que cette première régulation est correcte pour les autres approches débutant ainsi au point (sensibilité≃0,précision=1), celle classée par les RB est systématiquement fausse démarrant alors du point (sensibilité=0,précision=0). Cette situation est cependant vite corrigée avec la présence de vraies régulations dans la suite du classement. La méta-analyse reste globalement au dessus des trois approches sur lesquelles elle s'appuie, ce qui dénote une certaine cohérence entre elles. Seul le premier réseau du sous-challenge A300 montre une situation différente, dans ce cas les RB obtiennent un meilleur ratio précision/sensibilité, tandis que la méta-analyse se doit de faire le consensus des trois approches dont les classements semblent diverger plus fortement. Pour des populations moyennes (A300) les différences entre méthodes se resserrent ce qui tend à aplanir la majorité des tracés, ainsi la précision requise afin d'obtenir une sensibilité de 0.5, c'est-à-dire pour retrouver seulement la moitié des vrais arcs, est proche de 0. La situation continue d'empirer avec les plus petites populations (A100). Dans ce cas les méthodes obtiennent toutes de mauvais résultats, rendant difficile leur interprétation.

Figure 7.10 – **Courbes précision-sensibilité** pour le réseau le moins dense (à gauche) et le plus dense (à droite) de chaque sous-challenge. Chaque courbe représente les performances de l'une des 4 méthodes proposées : le sélecteur de Dantzig (en orange), les réseaux bayésiens (en bleu), la régression Lasso (en vert) et la méta-analyse (en rouge). L' arc de cercle en pointillés colorés représente la distance minimale atteinte par rapport au point (sensibilité=1,précision=1), tandis que sa couleur correspond à la méthode ayant atteint cette distance minimale. L'arc de cercle noir, représente cette même distance pour le graphe vide.



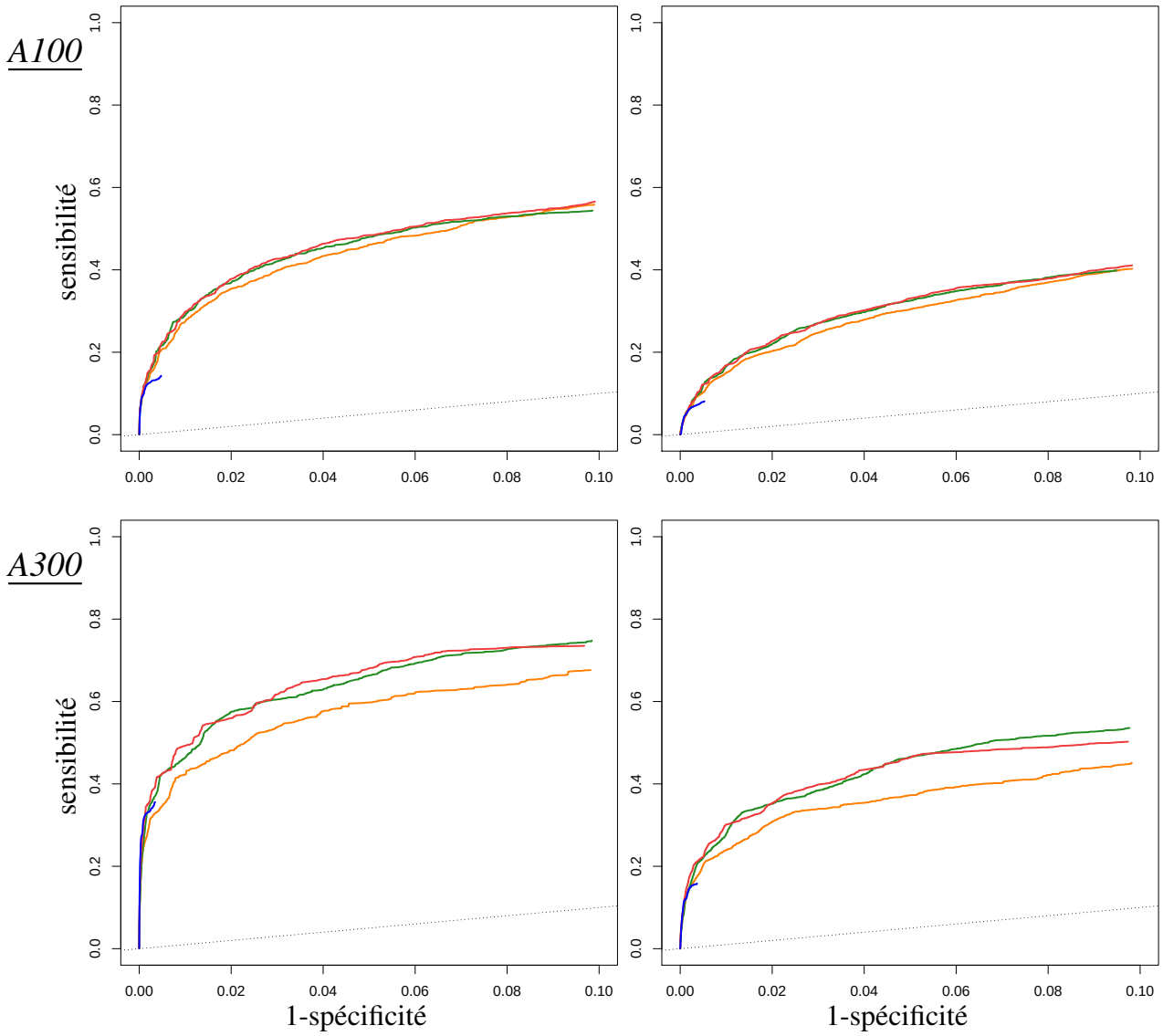


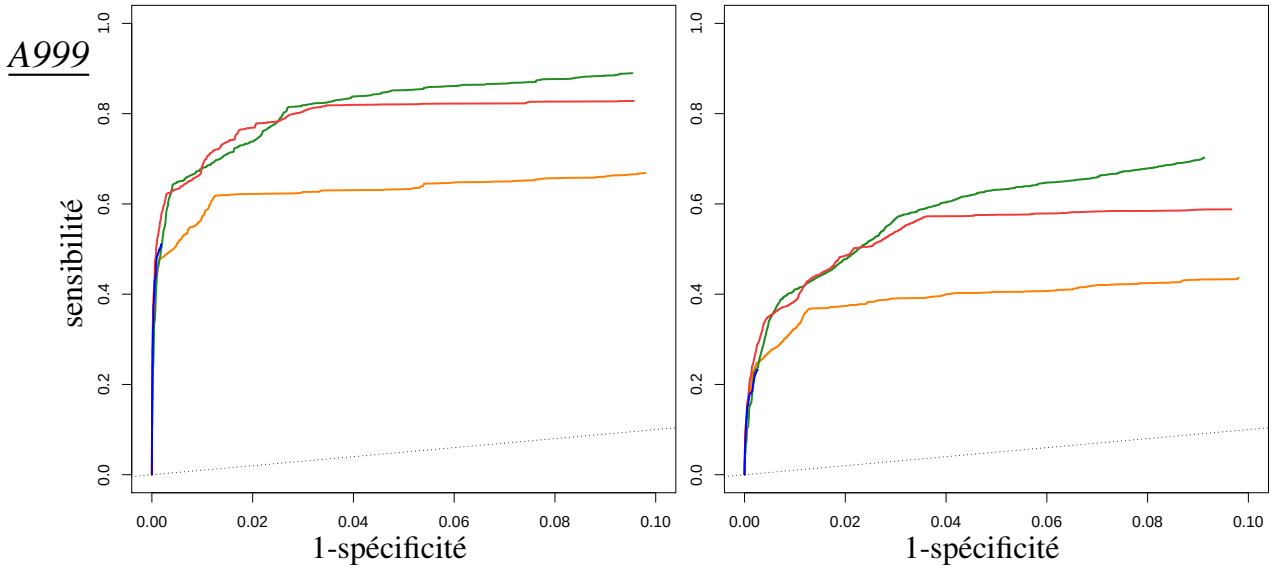
Courbes ROC Nous observons maintenant sur la Figure **7.11** les courbes ROC pour ces mêmes réseaux. A la différence des courbes précision-sensibilité, l'interprétation visuelle des courbes ROC dans le cas de réseaux peu denses est difficile. En effet dans ce cas, la spécificité qui mesure le nombre de non-régulations correctes par rapport au nombre de non-régulations du vrai réseau, reste très proche de 1 pour les premières régulations du classement. Ces premières régulations représentent pourtant l'information principale permettant de comparer les méthodes. Ainsi là où une courbe précision-sensibilité permet de distinguer visuellement la qualité de ces premières régulations au détriment d'une représentation compacte des dernières régulations du classement, les courbes ROC opèrent le principe inverse. La pente initiale de chaque courbe permet tout de même de mesurer la qualité de ces premières régulations. Dans le cas le plus favorable (premier réseau A999) la pente initiale forte indique ainsi une augmentation de la sensibilité tout en limitant le nombre de fausses régulations apprises, cette pente diminue d'autant que la complexité de l'apprentissage augmente (via le nombre d'individus et la connectivité du vrai réseau). Il est toutefois difficile de départager les méthodes uniquement d'après cette pente.

Le deuxième point de comparaison issu de la courbe ROC correspond au niveau de sensibilité atteint lorsque le taux de faux positif augmente. On observe alors que la régression Lasso retrouve un plus grand nombre de régulations, bien que celui-ci décroit lorsque la situation devient plus complexe. Suivant ces niveaux de sensibilité atteints la méta-analyse arrive seconde tandis que le sélecteur Dantzig arrive en troisième position. On observe toujours une réduction des écarts entre les méthodes lorsque le nombre d'individus se réduit. De la même manière que précédemment l'approche par RB est pénalisée lors du tracé de ces courbes, on peut ainsi voir pour certaines courbes l'arrêt prématuré à des taux de faux positifs très proches de 0 (pour le challenge A999 ces tracés, masqués par les autres méthodes, s'arrêtent respectivement de gauche à droite à des sensibilités de 0.5 et 0.23). La complétion de cette courbe, par tirage aléatoire des régulations manquantes, correspond alors à tracer, à partir de

ce point, une droite de pente identique à la ligne en pointillée noire. On imagine encore une fois les conséquences sur les aires mesurées.

Figure 7.11 – Courbes ROC pour le réseau le moins dense (à gauche) et le plus dense (à droite) de chaque sous-challenge. Chaque courbe représente les performances de l'une des 4 méthodes proposées : le sélecteur de Dantzig (en orange), les réseaux bayésiens (en bleu), la régression Lasso (en vert) et la méta-analyse (en rouge). La ligne en pointillée noire représente la courbe ROC d'un classement de 100 000 régulations tirées aléatoirement.





Étude de la méta-analyse Nous nous concentrons maintenant sur les performances de la méta-analyse pour le premier réseau du challenge A999. Nous observons sur la Figure **7.12**, la courbe précision-sensibilité dissociée en fonction du type réel de polymorphisme agissant sur chacun des deux gènes reliés. Cette distinction permet de repérer les effets des polymorphismes plus ou moins bénéfiques pour l'apprentissage de la régulation. Ainsi les régulations les mieux apprises sont celles pointant vers des gènes non soumis à une effet en *cis* de leur marqueur (tracés vert et rose). Ces perturbations dues aux effets *cis* s'expliquent du fait que dans ce cas chaque marqueur explique fortement son gène associé et occulte ainsi les autres régulations possibles. Cette perturbation est moins importante lorsque la régulation provient d'un gène ne possédant pas lui même d'effet *cis* (tracé bleu), la régulation s'effectue à la fois via le gène et le marqueur associé pesant ainsi davantage en tant que variables explicatives. Dans le cas où ce gène régulateur possède également un effet *cis* (tracé rouge), la régulation ne s'effectue qu'au travers du gène, rendant difficile sa détection.

Cohérences des approches La Figure **7.13** représente le diagramme de Venn indiquant les régulations communes parmi les 1 000 premières classées par chacune des approches (hors méta-analyse) pour le premier réseau du challenge A999. On remarque que près de la moitié de ces régulations sont communes entre les trois approches et qu'elles sont de plus correctes pour la majorité d'entre elles. Cela confirme la cohérence des prédictions pour ce réseau et donc les bonnes performances de la méta-analyse. La seconde plus grande intersection se situe entre la régression Lasso et le sélecteur de Dantzig, l'emploie pour ces deux méthodes d'un modèle linéaire explique vraisemblablement cette situation. Cependant le ratio de régulations correctes parmi cette intersection ne dépasse pas 40%. L'approche par RB s'offre quant à elle le poste d'outsider, les intersections avec les deux autres approches prises séparément étant plus limitées. Toutefois le nombre de vraies régulations apprises uniquement par le RB est supérieur au nombre appris, de manière séparée, par les approches basées sur le modèle linéaire. De même lorsque l'on inclue les intersections entre paire

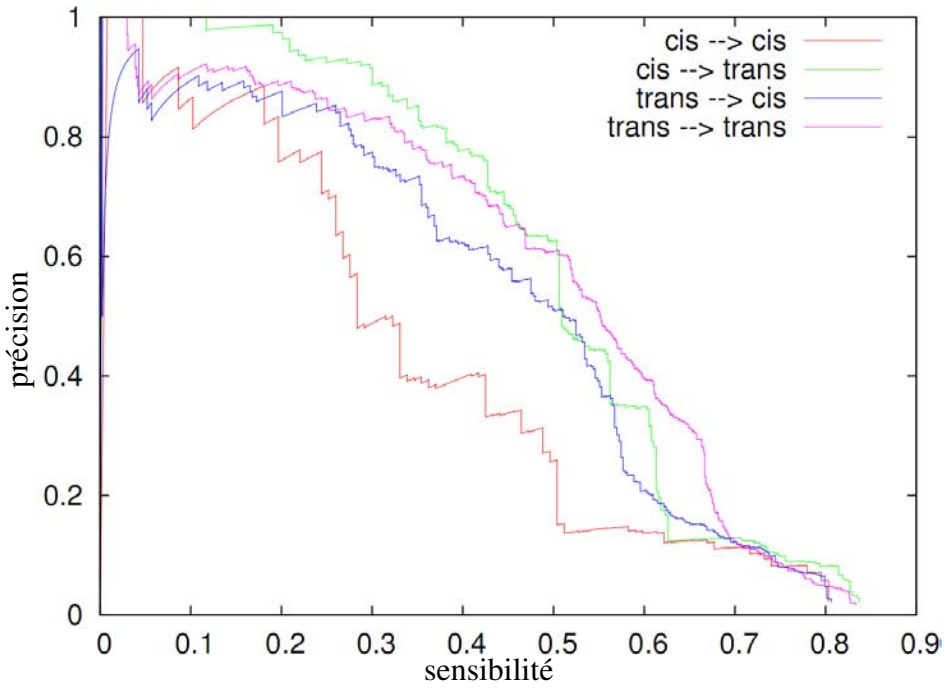


Figure 7.12 – Courbes précision-sensibilité de la méta-analyse pour le réseau le moins dense du sous-challenge A999. Les régulations sont divisées en 4 catégories en fonction de l'effet des marqueurs associés aux 2 gènes qu'elles relient. La distinction *cis*/*trans* pour un gène correspond donc au type du marqueur associé, c'est-à-dire à la position de celui-ci par rapport au gène.

d'approches. Le RB permet donc de détecter un plus grand nombre de régulations parmi les 1 000 premières classées, dû notamment à sa modélisation non-linéaire du phénomène de régulation.

Conclusion Nous venons de voir que notre approche basée sur les RB obtient de bon résultats pour les courbes précision-sensibilité où celle-ci est comparable voir supérieure dans certains cas au sélecteur de Dantzig. Ces performances sont cependant masquées par les critères utilisés pour comparer les méthodes durant ce challenge, pénalisant fortement le fait de ne fournir qu'un classement partiel des régulations. Ces critères sont d'autant plus critiquables dans le cas d'une application réelle à la reconstruction de réseaux de régulation, où seules quelques prédictions peuvent être validées expérimentalement. Ainsi prédire une liste de 100 000 régulations semble quelque peu inapproprié dans ce cas. Les résultats ont également mis en exergue l'intérêt d'effectuer une méta-analyse qui permet de tirer partie des différentes approches, cette méthode a par ailleurs obtenu les meilleurs résultats lors de ce challenge.

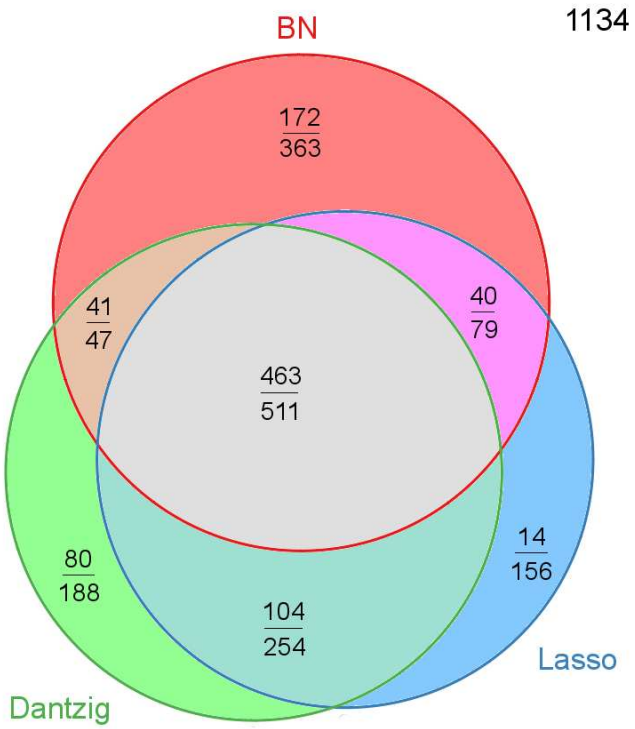


Figure 7.13 – Diagramme de Venn représentant le nombre de régulations en commun sur les 1 000 premières classées par les approches à base de réseaux bayésiens (en rouge), de régressions Lasso (en bleu) et du sélecteur de Dantzig (en vert). Chaque fraction représente le nombre de vraies régulations sur le nombre total de régulations communes. Le nombre de vraies régulations non-présentes dans l'union de ces approches s'élève à 1134.

7.2.2.6 Résultats supplémentaires

D'autres tests ont été réalisés sur ces mêmes données dans le cadre des RB et en dehors de la compétition DREAM, afin d'observer l'impact des nouveaux opérateurs présentés dans la partie 1 de ce manuscrit. Du fait que les autres méthodes comparées ici ne sont pas dédiées à l'analyse des données de *génétique-génomique*, nous n'utilisons que les données d'expression en omettant les variables marqueurs. Les réseaux à reconstruire sont donc composés de $p = 1000$ variables et possèdent une structure pouvant contenir des circuits. Cette situation représente ainsi un challenge pour la reconstruction de RB d'autant plus que le processus de génération des données diffère sensiblement de celui basé sur un ensemble de probabilités conditionnelles.

Nous revenons donc au cas d'une comparaison restreinte au cadre des RB, où chaque méthode est évaluée d'après les scores atteints ainsi que sur les critères structurels mesurés à partir des graphes obtenus non-orientés. Nous comparons la version la plus performante de SGS (à savoir SGS^3) utilisant le filtre défini en section 4.2, aux algorithmes GES évoluant dans l'espace des classes d'équivalence et MMHC qui effectue une recherche de type Hill-Climbing dans un espace restreint aux couvertures

de Markov de chaque variable. Nous utilisons ce dernier, dont une implémentation est proposée par Scutari [2010] dans le package R *bnlearn*, en remplacement de l'algorithme LAGD qui ne permet pas de traiter des réseaux de cette dimension. Nous analyserons également la qualité de notre filtre en le comparant à celui employé par MMHC. Ainsi seul l'algorithme GES ne subit aucune restriction de son espace de recherche. Cette différence de traitement s'explique par une implémentation optimisée de GES, lui permettant de traiter des réseaux de taille conséquente en un temps raisonnable sans avoir recours à un quelconque filtre.

Pour ces comparatifs, plus récents dans le déroulement de la thèse, nous utilisons l'implémentation en Comet pour l'algorithme SGS^3 ainsi que le score *BDeu* avec $\alpha = 1$. SGS^3 et MMHC sont exécutés 10 fois ($r = 10$) pour chacun des 15 réseaux puis nous conservons le meilleur score atteint lors de ces 10 répétitions. Tout comme LAGD, l'algorithme MMHC n'est pas stochastique par défaut, nous redéfinissons donc aléatoirement pour chaque répétition l'ordre des variables dans le fichier des données afin d'ajouter un caractère stochastique au choix des orientations.

Comparaison avec GES et MMHC Nous commençons par comparer les scores atteints par les trois approches. Le Tableau **7.2** présente ces résultats sous la forme d'un test de Wilcoxon, contrairement aux comparaisons effectuées sur les réseaux benchmark, nous disposons que d'un unique jeu de données pour chacun des réseaux. Afin d'évaluer malgré tout l'écart statistique entre les méthodes, nous considérons les résultats obtenus pour chaque sous-challenge comme des répétitions d'une même épreuve. Nous effectuons alors pour chaque sous-challenge un test de Wilcoxon portant sur les 5 réseaux pour une erreur de première espèce de 5% sans chercher à corriger le biais lié aux tests multiples.

Bien que la puissance de ce test est faible, on observe que SGS^3 atteint de meilleurs scores que ses concurrents pour toutes les tailles de population. On remarque également que GES se montre plus performant que MMHC dans le cas d'une petite population, allant à l'encontre des résultats observés lors des premières comparaisons où GES obtenait de mauvais résultats dans ces situations. Ce phénomène s'explique du fait que GES n'est pas restreint dans son espace de recherche à la différence de MMHC qui se voit fortement pénalisé par son filtre comme nous le détaillerons un peu plus bas.

Tableau 7.2 – Test de Wilcoxon (erreur fixée à 5%) pour les différents sous-challenges.

	A100	A300	A999
SGS^3 vs MMHC	+	+	+
SGS^3 vs GES	+	+	+
MMHC vs GES	-	~	+

La seconde comparaison s'effectue au niveau des critères structurels en débutant par les distances de Hamming présentées dans le Tableau **7.3**. Ces distances sont détaillées en fonction du nombre d'arêtes superflues et manquantes. On observe alors des performances médiocres pour toutes les approches, inférieures même à celles atteintes par la structure vide. Ces résultats montrent clairement la difficulté d'apprendre un réseau de cette dimension à partir de données non-générées d'après un RB.

On remarque globalement la faible densité des réseaux appris, conduisant à un grand nombre d'arêtes manquantes. SGS^3 apprend des réseaux plus denses mais qui possèdent de fait un nombre de *Faux Positifs* supérieur. Ainsi les meilleures approches d'après cette mesure correspondent généralement à celles apprenant les réseaux les moins denses. Il est alors difficile dans cette situation de comparer les méthodes. Seul le cas du sous-challenge A999 ne respecte pas cette règle où pour des distances de Hamming similaires, SGS^3 apprend un plus grand nombre d'arêtes que GES indiquant de fait de meilleures performances pour SGS^3.

Tableau 7.3 – Distance structurelle de Hamming (+ : arcs superflus ; - : arcs manquants) et nombre d'arcs appris en moyenne sur les 5 réseaux de chaque sous-challenge. Les meilleurs distances sont indiquées en gras.

Network	A100	A300	A999
SGS^3+	685	346	**429**
SGS^3-	3658	3476	**3365**
SGS^3(moyenne)	854	612	849
MMHC+	**276**	221	378
MMHC-	**3734**	3570	3459
MMHC(moyenne)	368	392	704
GES+	376	**234**	267
GES-	3718	**3549**	3530
GES(moyenne)	484	427	522

Afin d'éclairer ces résultats nous avons calculé pour chaque réseau les distances atteintes par rapport au point (sensibilité=0,précision=0), une valeur élevée indiquant alors une meilleure qualité de reconstruction. Ces valeurs sont représentées dans le Tableau **7.4**. On observe une disparité des résultats en fonction de chaque sous-challenge. Ainsi SGS^3 obtient les meilleurs résultats pour $n = 999$, GES lorsque $n = 300$ et enfin MCMC pour le plus petit nombre d'observations. Ce classement ne coïncide cependant pas avec les résultats observés suivant les scores atteints, où SGS^3 domine dans toutes les situations. Le nombre limité d'échantillons ainsi que la nature du processus ayant servi à les générer permettent d'expliquer cette différence. Dans ce cas le vrai réseau et le réseau optimal en terme de score divergent sensiblement, optimiser le score ne permet donc pas de se rapprocher systématiquement de la vraie structure. Ces deux réseaux convergent cependant lorsque la taille de la population augmente. Les résultats du sous-challenge A999 confirment cette idée où SGS^3 obtient à la fois les meilleurs scores et une qualité structurelle supérieure.

Comparatifs du filtre Nous nous attachons dans ce paragraphe à comparer la qualité de notre filtre avec la recherche des couvertures de Markov telle qu'elle est effectuée par MMHC. Nous présentons dans le Tableau **7.5** la taille et la sensibilité de l'ensemble des arcs potentiels pour le premier réseau

de chaque sous-challenge. La sensibilité des deux filtres est globalement assez faible, pour le cas favorable de la plus grande population notre filtre ne retient qu'un tiers des vraies régulations, ce taux descendant à moins de 10% lorsque $n = 100$. Notre filtre plus simple que la recherche des couvertures de Markov, garde néanmoins une meilleure sensibilité. Le nombre réduit d'arcs potentiels constitue à l'inverse un avantage permettant ainsi d'accélérer la recherche, on notera que les taux de réduction atteints dépassent systématiquement les 99% (sur les 999 000 arcs possibles).

Tableau 7.4 – Distances Euclidiennes à l'origine du point (précision, sensibilité) atteint. Les meilleures distances sont indiquées en gras.

		SGS[3]	MMHC	GES
A100	*Net1*	0.170	**0.218**	0.196
	Net2	0.213	**0.295**	0.232
	Net3	0.214	**0.266**	0.236
	Net4	0.201	**0.265**	0.214
	Net5	0.206	**0.247**	0.243
A300	*Net1*	**0.510**	0.483	0.464
	Net2	0.342	0.337	**0.385**
	Net3	0.484	0.488	**0.505**
	Net4	0.453	0.478	**0.498**
	Net5	0.419	0.397	**0.428**
A999	*Net1*	**0.578**	0.537	0.549
	Net2	**0.581**	0.510	0.505
	Net3	0.454	0.441	**0.484**
	Net4	**0.476**	0.450	**0.476**
	Net5	**0.479**	0.471	0.458

Tableau 7.5 – Nombre de parents potentiels retenus et sensibilité de cette sélection avec notre filtre (*BDeu* test) et celui utilisé dans MMHC pour le moins dense des réseaux de chaque sous-challenge

		A100	A300	A999
BDeu test	nombre retenu	2670	2430	5984
	sensibilité	9%	18%	35%
MMPC	nombre retenu	2568	3064	3842
	sensibilité	5%	12%	23%

7.3 Conclusion

Nous avons proposé dans ce chapitre deux modélisations dans le cadre des RB permettant de traiter des données de *génétique-génomique* afin d'améliorer l'apprentissage de la structure des RRG. La première d'entre elles est basée sur une modélisation séparée des variables d'expression et de polymorphisme et permet de représenter précisément les différentes situations de régulation. L'augmentation de l'espace de recherche due au nombre doublé de variables peut cependant être limité à l'aide de connaissances biologiques portant sur la zone de localisation des mutations au niveau des gènes. Le second modèle permet quant à lui de représenter de manière compacte le RRG mais perd en contre partie de sa puissance descriptive.

Nous avons par la suite présenté une série d'expérimentations portant sur les RRG artificiels *Web50*, afin de comparer les deux modélisations et de mesurer l'apport de l'information liée à la zone des mutations pour le modèle non-fusionné. Nous avons également analysé l'impact des critères étendus présentés dans la première partie de ce manuscrit. De cette première série nous avons pu conclure que le modèle non fusionné utilisant les différentes restrictions biologiques avec un *a priori* uniforme sur les classes de connectivité obtenait de meilleurs résultats. Nous avons donc comparer cette approche à un ensemble de méthodes représentatif de l'état de l'art dans le domaine de l'apprentissage de RRG. Au travers des diverses expérimentations menées sur ces données simulées avec des réseaux de taille modeste, nous avons noté les bon résultats de notre approche se classant parmi les méthodes les plus performantes dans la majorité des situations.

Cette approche a par ailleurs été reprise dans le cadre de la compétition internationale DREAM5 plus ambitieuse au niveau de la taille des réseaux de régulation et du nombre d'observations disponibles. Nous avons également présenté à cette occasion les autres méthodes proposées par divers membres de l'équipe SaAB dont une méta-analyse effectuant le consensus de ces approches. Cette dernière a par ailleurs obtenu les meilleurs résultats lors du challenge, démontrant ainsi l'intérêt de ce type d'approche similaire à une technique de *boosting*. Les comparaisons présentées ont permis de confirmer à nouveau le bon comportement des RB bien que pénalisés par la difficulté de notre approche à classer un nombre suffisant de régulations.

Quelques expérimentations supplémentaires sur ces données DREAM5, restreintes au cadre des RB, ont également permis d'observer l'écart entre l'objectif visant à maximiser une fonction de score et celui d'apprendre la vraie structure. Cette difficulté récurrente, liée aux problématiques de la grande dimension telles que l'apprentissage des RRG, amène à reconsidérer les score existants. C'est dans cette optique que nous avons proposé l'utilisation des scores étendus. Finalement nous avons validé le principe d'utiliser un filtre simple basé sur le même score que celui employé durant la recherche afin de restreindre l'espace des structures possibles. L'utilisation d'un filtre permet à la fois de considérer des réseaux de taille importante et d'accélérer la recherche. Il est toutefois nécessaire de définir des stratégies efficaces pour construire un tel filtre afin de réduire l'espace de recherche tout en limitant l'interdiction de vrais arcs.

Chapitre 8

Application aux données d'*Arabidopsis thaliana*

Sommaire

8.1 ***Arabidopsis thaliana*** **173**
8.2 **Analyse classique : recherche d'eQTLs** **174**
8.3 **Méthodologie employée** **174**
8.3.1 Complétions des données 174
8.3.2 Réglage de notre approche 175
8.4 **Résultats** **176**
8.4.1 Caractéristiques du réseau 176
8.4.2 Nomenclature des cas 178
8.4.3 Comptage des situations 180
8.4.4 Comparaisons et validation à partir de bases de connaissances biologiques . . 180
8.5 **Conclusion** **183**

Nous verrons dans ce chapitre, une application de notre méthode à des données réelles de *génétique-génomique* pour la plante *Arabidopsis thaliana*. Nous décrivons dans un premier temps les données utilisées et les prétraitements nécessaires à leur utilisation ainsi que les modifications spécifiques appliquées à notre méthode. Puis nous comparons le réseau appris à une recherche d'eQTLs avant de décrire deux situations cohérentes avec la littérature existante.

8.1 *Arabidopsis thaliana*

Arabidopsis thaliana également appelée Arabette, n'est pas qu'une jolie petite plante blanche. Elle fait partie à l'heure actuelle des plantes de référence dans le domaine de la génétique végétale du fait de son court cycle de vie (≈6 semaines) permettant ainsi de générer rapidement des populations et de sa capacité à s'auto-féconder. Son génome répartis sur 5 chromosomes comporte plus de 25 000 gènes, ce nombre augmentant régulièrement au fil des nouvelles découvertes.

Malgré la quantité importante d'observations biologiques disponibles sur *Arabidopsis thaliana*, les données de *génétique-génomique* sur des individus apparentés sont rares. Nous avons appliqué notre modèle non-fusionné à une population de 158 individus issus d'un croisement de type RIL [Simon et al., 2008]. Ces données contiennent pour chacun de ces individus les mesures d'expression obtenues grâce aux 32359 sondes d'une puce CATMA [Hilson et al., 2004], les génotypes relevés sur 89 marqueurs de type SNP répartis uniformément le long des 5 chromosomes d'*Arabidopsis*, ainsi que les positions physiques et génétiques des sondes et des marqueurs sur le génome. Les données brutes et normalisées sont disponibles dans la base de données CATdb [1][Gagnot et al., 2008].

Ce travail est issu d'une collaboration avec Olivier Loudet et Francisco Cubillos du centre INRA de Versailles et de Marie Laure Martin-Magniette de l'INRA d'Évry. Outre la mise à disposition de ces données, Olivier Loudet et Francisco Cubillos ont également effectué une recherche d'eQTLs sur celles-ci [Cubillos et al., 2012], nous utilisons cette analyse comme point de référence afin d'y comparer le réseau appris par notre approche.

8.2 Analyse classique : recherche d'eQTLs

Pour analyser nos résultats, nous nous comparons donc à l'analyse eQTL qui permet de définir pour chaque gène, un ou plusieurs marqueurs génétiques influençant le niveau d'expression du gène. Nous rappelons que cette analyse statistique établit des relations causales d'un marqueur vers un gène. Il est établi que la localisation de ces marqueurs causaux n'est pas exacte et que ces analyses ne s'attachent donc pas seulement au marqueur le plus explicatif localement mais délimitent des régions de confiance autour de lui. Si cette région inclut la position du gène nous parlerons d'eQTL de type *cis*, dans le cas contraire il s'agira d'un eQTL de type *trans*.

La recherche d'eQTLs sur ces données a permis d'établir pour 5035 sondes la présence d'un ou plusieurs eQTLs [2].

8.3 Méthodologie employée

Bien que les données utilisées ne soient pas brutes au sens où celles-ci ont été nettoyées des observations faussées et normalisées, quelques pré-traitements doivent encore être effectués avant d'appliquer notre méthode, notamment la complétion des données.

8.3.1 Complétions des données

Complétion des données d'expression Nous complétons dans un premier temps les données d'expression. Pour chaque sonde S_m présentant au moins un manquant nous sélectionnons un ensemble de 10 sondes S_N parmi celles ne comportant pas de manquants dont la covariance est la

1. http ://urgv.evry.inra.fr/CATdb/ ; Project : GNP07_RILKIT
2. http ://qtlstore.versailles.inra.fr/

plus forte avec la sonde à compléter. Puis nous calculons la valeur manquante D^i_m pour l'individu i par l'expression suivante :

$$D^i_{m/N} = \mu_{S_m} + Cov_{S_m,S_N} Cov^{-1}_{S_N} (D^i_{S_N} - \mu_{S_N})$$

où μ_{S_m} est la moyenne de D_m estimée sur l'ensemble des individus observés pour D_m, Cov_{S_m,S_N} est la covariance de la sonde S_m avec chacune des sondes de S_N, $Cov^{-1}_{S_N}$ est l'inverse de la covariance des sondes de S_N et $D^i_{S_N} - \mu_{S_N}$ l'écart pour chaque sonde de S_N entre la valeur mesurée pour l'individu i et la moyenne sur l'ensemble des individus.

Une fois ces données complétées, nous filtrons à deux reprises les sondes que nous souhaitons conserver dans notre modèle. La première sélection consiste à ne garder que les sondes dont au moins un eQTL a été détecté avec un score LOD supérieur à 2.5 par l'analyse eQTL. Notre but est ici de définir les informations supplémentaires apportées par notre approche par rapport à l'analyse eQTL. Ce premier filtre réduit le nombre de sondes de 32359 à 4177. Pour le second filtre nous recherchons les groupes de sondes ayant une corrélation de Pearson supérieur à 0.95 afin de ne conserver qu'un seul représentant de chacun de ces groupes. Nous savons que des variables fortement corrélées seront très vraisemblablement reliées dans le réseau appris et agiront comme une unique variable. Nous préférons donc réduire ce nombre de variables en double dans notre modèle afin de diminuer la complexité de la recherche. Cependant ce dernier filtre ne fait ressortir que 2 sondes fortement corrélées réduisant ainsi à 4176 le nombre final de sondes composant notre analyse.

Nous avons finalement discrétisé ces données avec la méthode adaptative présentée dans la section 4.1.1.

Complétion des données de génotype Les données de génotypes comportant également certaines données manquantes, nous utilisons le package R *qtl* afin de les compléter et de créer par la même occasion des $pseudo-marqueurs$ espacés d'1 cM entre chaque vrai marqueur afin de couvrir plus finement la carte génétique. Les valeurs de ces $pseudo-marqueurs$ pour les 158 individus sont les probabilités d'observer l'un des deux allèles, sachant les informations de génotypage. Ces probabilités sont discrétisées en 3 états en utilisant les seuils (0.75 et 0.25). Un $pseudo-marqueur$ de classe 1 indique une probabilité supérieure à 0.75 d'observer l'allèle de référence (et donc inférieur à 0.25 d'observer l'allèle muté) tandis qu'une classe 2 indique une probabilité d'observer l'allèle de référence comprise entre 0.75 et 0.25. Cet état intermédiaire représente donc l'incertitude sur les $pseudo-marqueurs$ lorsque les deux vrais marqueurs flanquants informatifs n'indiquent pas le même allèle. La classe 3 représente logiquement une probabilité supérieure à 0.75 d'observer l'allèle muté.

Nous obtenons au total 590 marqueurs incluant les 89 vrais marqueurs initiaux complétés.

8.3.2 Réglage de notre approche

Outre une augmentation du nombre de variables et une réduction du nombre d'individus, la principale différence entre ces données réelles et celles employées dans les sections précédentes, provient du nombre de marqueurs inférieur au nombre de gènes. Bien que notre modélisation

du problème soit initialement conçue pour des données présentant le même nombre de gènes que de marqueurs, il est possible d'adapter celui-ci au cas présent. Nous n'associons donc plus systématiquement un marqueur à chaque gène. Ainsi nous autorisons des situations où un marqueur se situe dans une région inter-génique ou bien qu'un gène ne possède aucun marqueur à proximité sur le génome.

Il est toutefois possible d'utiliser de la même manière des connaissances sur la position des marqueurs afin de fixer ou d'interdire certaines relations, que ce soit pour des eQTLs agissant en *cis* ou encore ceux ayant un effet *trans* et dont la position coïncide avec celle d'un gène qui ne subit pas d'effet *cis* de sa part. Cette dernière situation peut alors s'interpréter comme la présence d'une mutation dans la région codante du gène. Cependant, nous n'utilisons pas cette information sur les effets des marqueurs issue de l'analyse eQTL du fait que cette dernière est utilisée comme point de comparaison, ainsi pour notre méthode seules les restrictions interdisant la présence d'arcs pointant vers les marqueurs sont appliquées.

De plus indépendamment de la recherche d'eQTLs nous avons utilisé le filtre présenté dans le section 4.2 qui sélectionne pour une variable cible tout parent qui permet à lui seul d'améliorer le score local à la cible par rapport à l'absence de parent. Nous avons donc pour chaque variable d'expression une liste de parents potentiels. Nous regarderons brièvement la restriction effectuée par ce filtre dans la section 8.4

L'apprentissage de la structure est effectué grâce à l'algorithme SGS[1] sans répétition ($r = 1$) dû au temps de calcul nécessaire pour fournir un seul réseau ($\approx 8^h$) et en utilisant le score *BDeu* dont le paramètre $\alpha = 1$. Nous fixons le nombre maximum de parents à 4. Afin d'augmenter la précision de la reconstruction nous avons fixé le paramètre du score *BDeu* étendu $\gamma = 1$ (voir section 4.3), ce réglage permet par ailleurs de réduire à nouveau la durée de la recherche du fait de l'effet de ce paramètre sur la densité du réseau appris.

8.4 Résultats

8.4.1 Caractéristiques du réseau

Le réseau appris par GS est peu dense, 6137 arcs relient 4766 variables soit une densité d'environ 1.3. Le réseau est constitué d'une composante connexe majeure regroupant 4004 variables, les autres variables forment des composantes de dimension inférieure à 5. Seules 284 relations sont de type $M \rightarrow E$ tandis que les 5853 autres relations représentent des régulations entre variables d'expression, un nombre assez faible au vu du nombre d'eQTLs détectés.

L'analyse des degrés entrants et sortants du réseau appris (Tableau **8.1**), montre que 278 sondes possèdent 3 parents ou plus (maximum de 4 parents atteint pour 41 sondes) tandis que d'autres diffusent fortement allant jusqu'à des degrés sortants de 55.

Tableau 8.1 – Nombre de variables d'expression et de marqueur suivant leurs degrés entrant et sortant.

Degré	0	1	2	3	4	5	6	7	8	9	10+
Degré entrant des gènes	263	2008	1627	237	41	-	-	-	-	-	-
Degré sortant des gènes	2164	844	465	260	118	98	58	46	26	29	68
Degré sortant des marqueurs	457	67	29	15	9	7	2	1	3	-	-

Nous pouvons voir sur l'histogramme de la Figure **8.1**, l'effet de la pré-sélection sur le nombre de régulateurs potentiels, ce même histogramme avant la pré-sélection ne comporterait qu'une seule classe où toutes les variables d'expression posséderaient 4765 parents potentiels (4176+590-1). Cette pré-sélection réduit donc fortement l'espace de recherche et interdit même totalement l'ajout de parents pour 255 sondes.

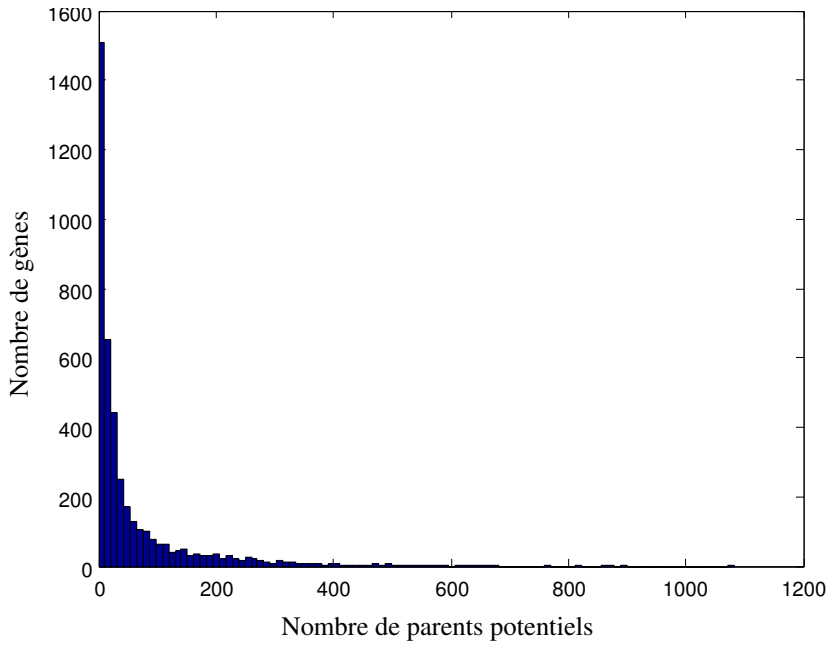


Figure 8.1 – Histogramme du nombre de parents potentiels lors de la recherche pour les variables expressions.

8.4.2 Nomenclature des cas

Afin d'analyser ce réseau, nous étudions successivement chaque eQTL de l'ensemble proposé par O. Loudet. Pour chacun de ces eQTLs nous étudions les configurations possibles dans le RB permettant de l'expliquer.

Les Figures **8.2**, **8.3**, **8.4**, **8.5**, **8.6** et **8.7** répertorient les différentes configurations possibles permettant d'expliquer un eQTL au travers du RB. Chacune de ces figures représente sur deux niveaux les variables marqueurs et d'expression. Les eQTLs détectés sont représentés par une zone rouge (intervalle de confiance) s'étendant sur plusieurs marqueurs ayant une influence sur l'expression de la sonde E_1. Nous distinguons dans les Figures **8.2**, **8.3** et **8.4** des situations où l'eQTL a été détecté en *cis* tandis que les Figures **8.5**, **8.6** et **8.7** représentent des eQTLs en *trans*. En bleu est représentée la configuration apprise dans le RB, lorsque qu'une de ces configurations se présente, nous parlerons d'eQTL "expliqué" par le RB. Certaines configurations pouvant s'inclure (exemple les cas 2 et 3 ou 5 et 6) nous définissons un ordre de priorité $cas\ 1 > cas\ 2 > cas\ 3$ et $cas\ 4 > cas\ 5 > cas\ 6$ afin de ne compter qu'une seule explication par eQTL. Cet ordre ne représente cependant pas un classement jugeant de la vraisemblance de la configuration par rapport au réseau réel mais est utilisé uniquement dans un but de comparaison avec l'analyse eQTL.

Les cas 1 et 4 coïncident fortement avec l'analyse eQTL en indiquant qu'un marqueur agit directement sur l'expression d'une sonde. Les cas 2 et 5 indiquent quant à eux que cette régulation est indirecte et passe par une seule ou une succession de sondes. Ce *chemin* débute malgré tout par une relation de type $M \rightarrow E$. Point de départ que nous ne retrouvons pas dans les cas 3 et 6. Dans ces deux derniers cas, seule la présence d'une sonde régulant (directement ou non) notre sonde E_1 et se situant physiquement dans l'intervalle de confiance de l'eQTL fournit une explication. Cette configuration représente le cas d'un marqueur responsable de la variation d'expression de E_1 situé probablement dans la zone codante d'un gène régulateur de E_1, empêchant donc la détection de l'eQTL (en *cis*) pour ce gène régulateur.

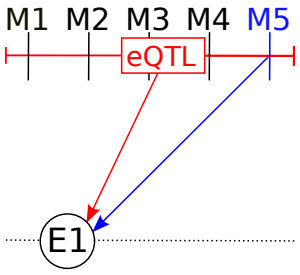


Figure 8.2 – Cas 1 : Situation où l'algorithme apprend comme parent direct un marqueur situé dans l'intervalle de confiance de l'eQTL *cis* détecté.

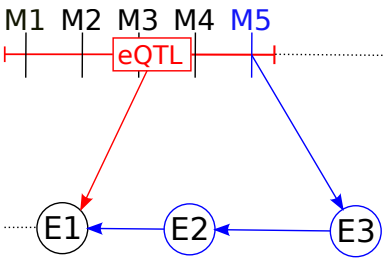


Figure 8.3 – Cas 2 : Situation où le réseau bayésien contient un chemin composé de sondes débutant par un marqueur situé dans l'intervalle de confiance de l'eQTL *cis* détecté.

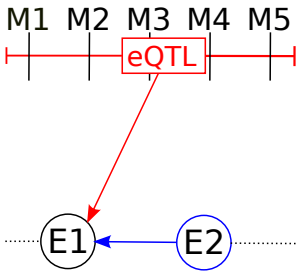


Figure 8.4 – Cas 3 : Situation où le réseau bayésien contient un chemin composé de sondes débutant par une sonde située physiquement dans l'intervalle de confiance de l'eQTL *cis* détecté.

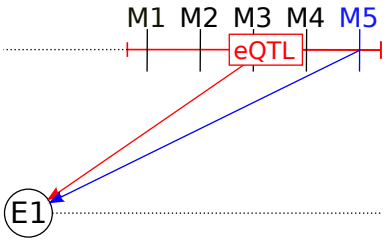


Figure 8.5 – Cas 4 : Situation où l'algorithme apprend comme parent direct un marqueur situé dans l'intervalle de confiance de l'eQTL *trans* détecté.

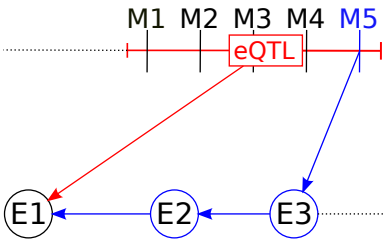


Figure 8.6 – Cas 5 : Situation où le réseau bayésien contient un chemin composé de sondes débutant par un marqueur situé dans l'intervalle de confiance de l'eQTL *trans* détecté.

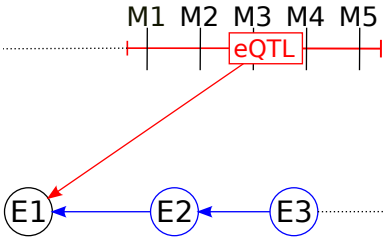


Figure 8.7 – Cas 6 : Situation où le réseau bayésien contient un chemin composé de sondes débutant par une sonde située physiquement dans l'intervalle de confiance de l'eQTL *trans* détecté.

Nous définissons une marge d'erreur admissible permettant d'élargir de part et d'autre l'intervalle de confiance de l'eQTL afin de comprendre l'origine possible des eQTLs non expliqués. Nous avons fait varier cette marge de 0 à 5 cM.

8.4.3 Comptage des situations

Sur les 6453 eQTLs détectés lors de l'analyse faite par O. Loudet, nous nous intéressons qu'aux eQTLs portant sur les 4176 sondes filtrés précédemment (soit 5100 eQTLs) puis nous nous restreignons aux 1269 ayant le plus haut scores LOD, correspondant à une sélection pour un FDR de 1%. Ces eQTLs ont été détectés pour 1219 sondes différentes, et se répartissent de la manière suivante : 938 eQTLs en *cis* / 331 eQTLs en *trans*. Cette sur-représentation des eQTLs en *cis* s'explique par la nature plus forte de ce type d'effet sur l'expression des gènes.

Le Tableau **8.2** représente le nombre d'eQTLs validés en fonction des différentes explications possibles lorsque nous faisons évoluer la marge d'erreur de 0 jusqu'à 5 cM.

Marge	Cas 1	Cas 2	Cas 3	**Tot-Cis**	Cas 4	Cas 5	Cas 6	**Tot-Trans**
0 cM	156	342	325	**823**	43	118	106	**267**
1 cM	187	401	277	**865**	58	148	88	**294**
2 cM	202	435	243	**880**	68	161	69	**298**
3 cM	209	459	221	**889**	68	167	67	**302**
4 cM	212	475	208	**895**	69	170	65	**304**
5 cM	212	490	196	**898**	69	173	63	**305**

Tableau 8.2 – Nombre d'eQTLs validés suivant les différentes configurations possibles pour une marge d'erreur variant de 0 à 5 cM

Au vu des variations observées nous pouvons garder une marge de 2cM de part et d'autre afin de garder un compromis acceptable entre le nombre d'eQTLs validés et la précision de l'intervalle de confiance.

8.4.4 Comparaisons et validation à partir de bases de connaissances biologiques

Nous pouvons voir sur le Tableau **8.2** que le RB permet d'expliquer respectivement 93% et 90% des eQTLs détectés de type *cis* et *trans*. En considérant uniquement les situations où le chemin détecté remonte jusqu'à un marqueur (cas 1, 2, 4 et 5) ces valeurs descendent à 67% et 69% parmi lesquelles plus de 68% sont expliquées de manière indirecte dans le réseau. Le RB propose ainsi un nombre plus élevé d'explications impliquant des régulations entre gènes que d'interactions directes *marqueur-gène*. Cette observation justifie alors l'intérêt d'un modèle non-fusionné qui permet de saisir au mieux la complexité des interactions.

Nous sommes cependant conscients de la faiblesse de certaines situations, notamment des cas 2 et 3 du fait que l'effet *cis* détecté est très certainement direct. Ainsi l'explication proposée de cet eQTL via des gènes pouvant être éloignés de l'eQTL (dans le cas 2) semble peu probable, du moins en terme d'orientation des régulations.

Afin d'étudier les plus fortes relations de notre réseau, nous classons les arcs appris en fonction de l'amélioration produite en terme de score *BDeu* lors de leur ajout dans le réseau par l'algorithme GS. Puis nous nous intéressons aux seules relations entre gènes situés sur des chromosomes différents afin d'éviter tout biais d'interprétation dû à leur proximité. Enfin pour les 10 premières régulations, nous effectuons une recherche de *gene ontology* commune entre chaque couple de gènes reliés, d'après la base de données *TAIR* [Swarbreck et al., 2008]. Ces *gene ontology* qui permettent de standardiser les connaissances sur les gènes et leurs produits, regroupent également les informations portant sur les fonctions connues des gènes au sein de la cellule.

Ainsi parmi les arcs présents dans notre réseau, nous observons un arc entre les gènes *LHCB6* (*AT*1*G*15820) et *ACP4* (*AT*4*G*25050) situés respectivement sur les chromosomes 1 et 4. Pour chacun de ces deux gènes un eQTL est détecté sur le chromosome 1, à proximité justement de LHCB6. Cette situation est représentée sur la Figure **8.8** (a). Nous sommes alors dans une situation où un même eQTL est détecté en tant que *cis* pour le gène *LHCB6* et en *trans* pour le gène *ACP4*, bien que dans ce cas, aucun arc impliquant un marqueur n'a été appris par le RB. Cette différence peut s'expliquer par le score pénalisé qui est utilisé et qui n'autorise que peu de parents pour chaque variable comme le montre la faible connectivité du graphe. Notre méthode établi un lien direct entre ces deux gènes connus pour participer tous deux aux processus de réponse à la lumière chez *Arabidopsis*, en effet Anderssona et al. [2001], Bonaventure and Ohlrogge [2002] ont montré que l'expression du gène *ACP4* augmente fortement en présence de lumière tandis que le gène *LHCB6* intervient dans l'activité de photo-synthèse de la plante.

On peut visualiser ce lien sur la Figure **8.8** (b) à l'aide du logiciel *Pathway Studio* [Nikitin et al., 2003] qui permet de représenter sous forme de graphe, des connaissances extraites de la littérature. Ainsi les liens tracés entre les différents produits ou fonctions cellulaires peuvent être issus de simples co-citations dans les mêmes articles, ou de connaissances plus précises sur le rôle identifié de certains gènes. Dans notre cas les deux gènes *LHCB6* et *ACP4* sont bien connectés à la même fonction cellulaire lié à la photosynthèse.

Cette exemple pose également le problème de la causalité, trop peu abordée durant cette thèse. Comme nous le remarquons, l'arc appris est orienté de *ACP4* vers *LHCB6* alors même que ce dernier est régulé en *cis*. Dans ce cas, il est probable que l'eQTL ait en réalité un effet en *cis-trans* sur *ACP4* via *LHCB6*, l'arc appris serait alors mal orienté. Cette erreur peut être due au phénomène d'orientations Markov-équivalentes, présenté dans la première partie de ce manuscrit. Ainsi, une fois cet arc ajouté au réseau avec la mauvaise orientation, l'algorithme a été incapable d'inverser celui-ci. Cet arc ne possède donc aucun caractère causal, il reste cependant pertinent et son orientation peut être retrouvée grâce à des informations supplémentaires telles que la présence d'eQTLs.

Une autre situation particulièrement intéressante, présentée sur la Figure **8.9** (a), est la présence dans notre réseau d'un arc entre les gènes *LURP1* (*AT*2*G*14560) et *WRKY70* (*AT*3*G*56400) situés respectivement sur les chromosomes 2 et 3 et pour lesquels un eQTL commun est détecté sur le chromosome 4. Chacun de ces gènes est connu pour participer au système de défense de la plante contre des champignons [Knoth et al., 2007] ce qui justifie pleinement la présence de cet arc dans notre

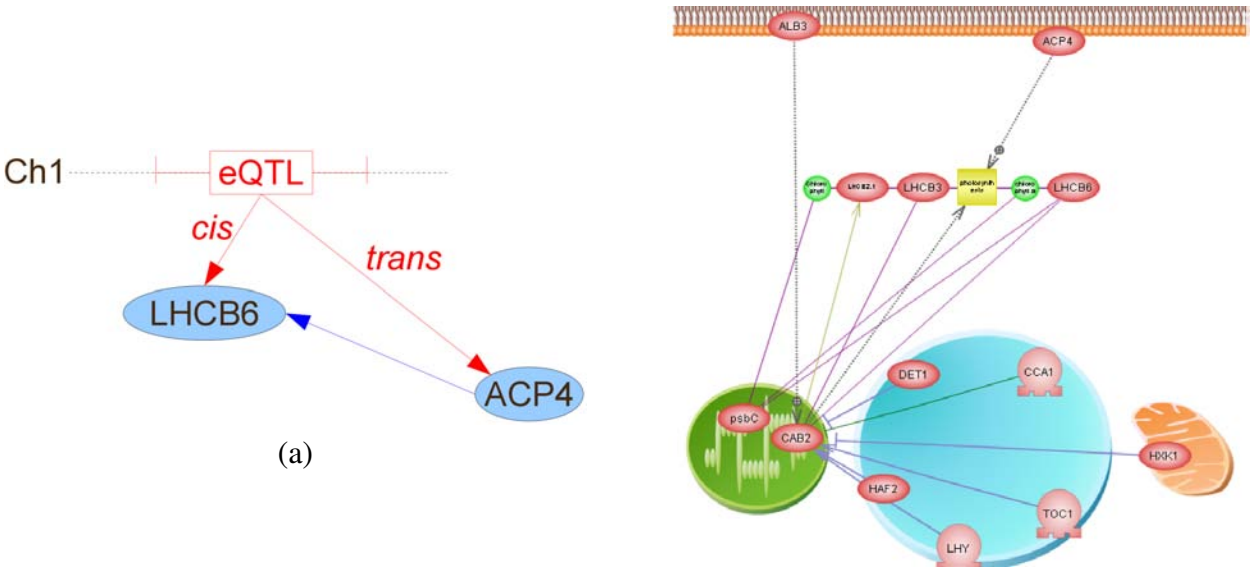


Figure 8.8 – (a) Régulation apprise (en bleu) par le réseau bayésien entre les gènes *LHCB6* et *ACP4* pour lesquels un eQTL commun (en rouge) a été détecté sur le chromosome 1. (b) Vue à l'aide du logiciel *Pathway Studio* du voisinage de chacun de ces deux gènes suivant les références bibliographiques communes.

réseau. Contrairement au cas précédent aucun de ces deux gènes n'est détecté en *cis* car leur eQTL commun se situe sur un autre chromosome, ces deux relations sont donc de type *trans*. Pour autant aucune autre relation directe ne lie un gène situé sur le chromosome 4 à ces deux gènes dans notre réseau. La raison de cette absence s'obtient en analysant dans la littérature les autres gènes impliqués dans les processus de défense d'*Arabidopsis*. On observe ainsi que le gène *RPP5* (*AT*4*G*16950) situé sur le chromosome 4 fait parti de ce processus et se situe de plus dans la région de confiance de notre eQTL commun. *RPP5* correspond donc très vraisemblablement au gène permettant d'établir le lien entre l'eQTL et les gènes *LURP1* et *WRKY70*. Cependant ce gène n'est pas mesuré par la puce CATMA et il n'est donc pas présent dans le réseau appris. Dans cette situation l'arc appris par notre méthode peut être interprété de deux manières. La première correspond à une situation réelle divergente où les deux gènes sont frères dans le vrai réseau et sont donc reliés en l'absence de leur père. La seconde représente une liaison en série dans le vrai réseau où le gène non-mesuré sur le chromosome 4 régule directement l'un des deux gènes et indirectement le second. Dans ce dernier cas, l'arc appris représente une vraie régulation directe.

En regardant cette fois la Figure **8.9** (b) qui détaille la vue obtenue à partir du logiciel *Pathway Studio* on retrouve effectivement les deux gènes *LURP1* et *WRKY70* gravitant autour de la même fonction de défense à la maladie. De plus ces gènes sont reliés par un arc indiquant la présence dans la littérature d'une régulation observée de *WRKY70* vers *LURP1*, confirmant ainsi la thèse d'une liaison en série des deux gènes. On repère également le gène *RPP5* relié à *LURP1* ainsi qu'à cette même fonction.

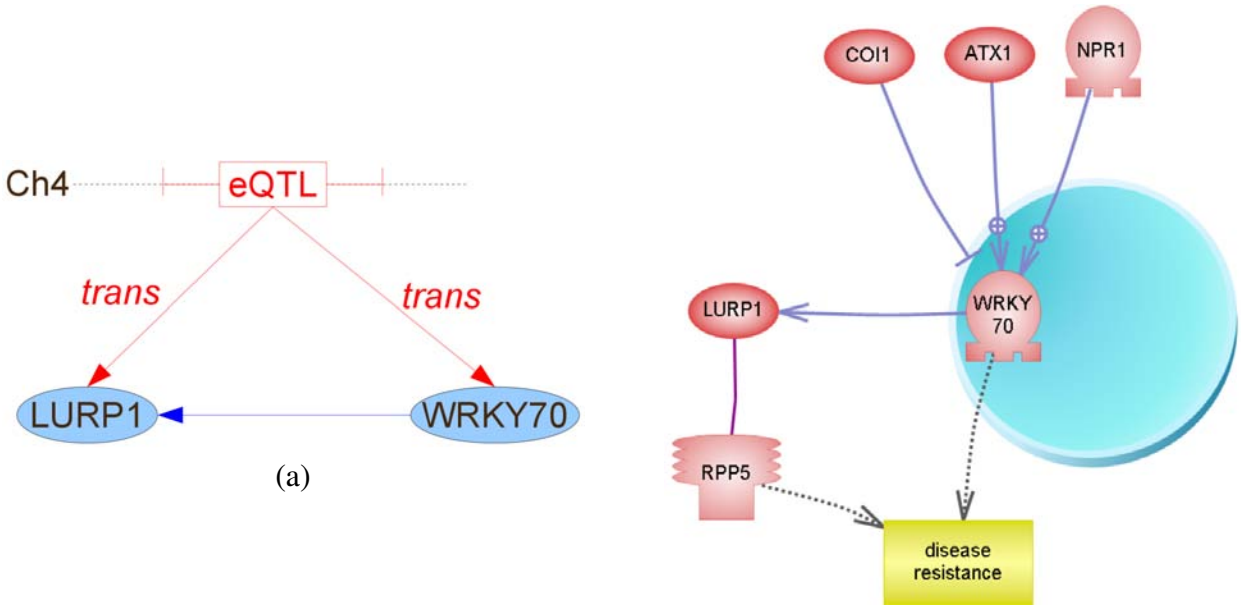


Figure 8.9 – (a) Régulation apprise (en bleu) par le réseau bayésien entre les gènes *LURP1* et *WRKY70* pour lesquels un eQTL commun (en rouge) a été détecté sur le chromosome 4. (b) Vue à l'aide du logiciel *Pathway Studio* du voisinage de chacun de ces deux gènes suivant les références bibliographiques communes.

Nous pouvons observer dans ces deux exemples qu'aucune relation d'un marqueur vers un gène n'est apprise par notre méthode malgré la pertinence de ces arcs notamment dans la dernière situation où un marqueur proche de l'eQTL détecté sur le chromosome 4 pourrait logiquement apparaître en tant que parent des deux gènes *LURP1* et *WRKY70*. L'explication peut provenir du faible score LOD de l'eQTL détecté dans ce cas ($\approx$ 3.5) ainsi que du choix des seuils de discrétisation utilisés pour la définition des pseudo marqueurs. D'après le réseau appris, les régulations directes entre gènes semblent donc prévaloir sur les influences en *trans* des marqueurs sur l'expression des gènes. Il n'est cependant pas clair de savoir si ce phénomène est dû à un fait biologique réel ou à un biais dans le pré-traitement des données.

8.5 Conclusion

Nous avons présenté dans ce chapitre des premiers résultats obtenus à partir de données réelles de *génétique-génomique* pour *Arabidopsis thaliana*. Le nombre important de gènes mesurés pour un faible nombre d'individus a nécessité de filtrer fortement les gènes à inclure dans le modèle. Nous avons également dû compléter les données manquantes et créer des *pseudo-marqueurs* afin de couvrir plus finement le génome. A la différence des expérimentations précédentes, nous n'avons pas utilisé ici l'information sur la zone de localisation des marqueurs, obtenue à l'aide d'une recherche d'eQTLs. Cette analyse a au contraire servi de point de référence afin d'étudier le réseau appris par

notre approche.

Ce réseau possède une densité faible et présente une sur-représentation de régulations directes entre gènes. Nous avons montré que notre prédiction était à la fois cohérente avec l'analyse eQTL et qu'elle proposait des chemins permettant d'expliquer certaines relations de type *trans*. Nous avons par ailleurs validé certaines de ces régulations à partir de la littérature. Ces travaux représentent un point de départ à une analyse approfondie de ce réseau ainsi qu'à l'apprentissage de réseaux incluant plus de gènes.

Perspectives pour l'application aux données biologiques

Dans cette seconde partie nous avons présenté une application de la reconstruction des réseaux bayésiens à l'apprentissage des réseaux de régulation de gènes. Dans cette optique nous avons proposé deux modélisations possibles permettant de représenter des données d'expression et de polymorphismes. L'un de ces modèles a par la suite été comparé à de nombreuses autres méthodes de reconstruction dans différentes situations. L'intérêt d'une modélisation cohérente des deux types de données et d'une utilisation appropriée des informations portées par les polymorphismes a ainsi pu être validé. Ce travail peut être étendu suivant plusieurs directions. La première d'entre elles, populaire actuellement, consiste à intégrer d'autres sources de données, ces informations supplémentaires pouvant provenir des sites de fixation, des régulations entre protéines ou bien de la littérature.

Cette approche nécessite cependant de définir un modèle plus souple permettant d'intégrer la variabilité de ces données. Le modèle proposé devra donc effectuer un compromis entre sa puissance descriptive et sa complexité. Il faudra ainsi éviter la fusion systématique des données au risque de réduire fortement leur information propre, sans pour autant représenter chaque acteur biologique par une variable distincte dans le modèle. Cette intégration au sein d'un même réseau bayésien, bien qu'intéressante, nécessite une réflexion approfondie sur le modèle à employer qui doit rester à la fois cohérent et applicable aux données disponibles.

Un second élément soulevé durant cette application relève de la dimension des réseaux à reconstruire et de la nécessité d'utiliser un filtre. Nous avons vu que le filtre proposé permet de réduire fortement l'espace de recherche, autorisant ainsi l'exécution de notre algorithme sur des réseaux de tailles conséquentes. Cependant la sensibilité des arcs conservés par ce filtre est faible. Des travaux doivent donc être menés afin de définir une nouvelle stratégie permettant de conserver davantage de vraies régulations sans pour autant augmenter trop fortement la taille globale du filtre.

Nous avons également présenté à la fin de cette seconde partie, les premiers résultats obtenus à partir de données réelles issues de la plante *Arabidopsis thaliana*. Un premier réseau reconstruit a ainsi pu être analysé et comparé avec une approche classique afin de s'assurer notamment de leur cohérence. Quelques relations apprises ont également été validées par la littérature, cependant cette première étude ne peut être que les prémices d'une analyse approfondie de ces données. La première voie nécessite d'apprendre plusieurs de ces réseaux. Nous n'avons pas eu le temps jusqu'à présent de répéter l'algorithme SGS, il serait toutefois intéressant d'observer la variabilité des réseaux appris suite à ces répétitions ou via l'utilisation d'une technique de *boostrap*. Une liste de relations robustes

pourrait alors être établie afin que certaines d'entre elles puissent être validées expérimentalement. L'autre voie consiste à étudier plus précisément ce premier réseau appris en regardant par exemple les régulations touchant à un ensemble de gènes d'intérêt afin de valider celles-ci par la littérature. De la même manière nous pouvons nous concentrer sur des configurations singulières de ce réseau telles que les *hubs*, afin de rechercher si ces gènes ont effectivement un rôle reconnu d'après la littérature.

Une seconde perspective à court terme pour ces données réelles, consiste à intégrer l'information partielle des effets des polymorphismes. En effet, afin de ne pas biaiser la comparaison avec l'analyse eQTL, nous n'avons pas utilisé lors de la reconstruction l'information liée aux effets *cis* et *trans* des marqueurs qui permet pourtant de restreindre judicieusement le modèle. Il est de fait imaginable d'effectuer ces restrictions, mêmes partielles, dans le réseau et d'observer les changements de la structure apprise.

Nous avons également évoqué le problème de la causalité des relations apprises dans les réseaux bayésiens. Nous savons qu'un arc appris ne représente aucune causalité dès lors que celui-ci est inversible au sens des équivalents de Markov, cependant les polymorphismes peuvent orienter fortement certaines régulations. Dans cet objectif, nous avons déjà appliqué certaines restrictions à notre modèle issues des polymorphismes, cependant nous pouvons aller plus loin dans leur utilisation. L'une des perspectives porte alors sur l'utilisation des données de polymorphismes afin d'orienter les arcs inversibles entre les gènes. Cette étape peut s'effectuer en amont de la recherche et être utilisée sous la forme d'un *a priori* de façon similaire à Zhu et al. [2007] ou durant la recherche à l'aide d'un algorithme spécifique utilisant cette information pour chaque choix équivalent d'orientation. Nous pouvons également penser à une correction *a posteriori* du réseau par les polymorphismes.

La dernière perspective revêt cette fois un aspect plus technique et consiste en une amélioration de l'implémentation actuelle de nos algorithmes, implémentation qui ne permet pas en l'état actuel de traiter des réseaux comportant plus de 4 000 variables. Un changement de langage apparaît alors nécessaire. De plus une réflexion sur les structures de données à employer doit être engagée afin de maintenir une information suffisamment condensée et ne nécessitant que peu de re-calculs après chaque modification du graphe.

Remarques générales

Nous avons vu dans cette thèse la difficulté éprouvée afin d'apprendre un réseau de régulation de gènes aussi bien que celle liée à l'apprentissage d'un réseau bayésien générique. Le faible nombre d'observations lié au nombre élevé de variables semble bien insurmontable pour toutes les approches actuelles. L'une des voies prometteuse pour l'apprentissage de réseau de régulation reste principalement l'intégration de données variées, permettant alors d'augmenter la robustesse des prévisions. Cependant même dans ces conditions, la reconstruction du réseau complet semble encore hors de portée.

De plus intégrer des types de données variées s'accompagne bien souvent d'une augmentation de la variabilité des provenances de celles-ci. En effet obtenir l'ensemble de ces données pour une même population représente un protocole expérimental lourd, pour cette raison certaines approches utilisent des données issues de différentes expérimentations. L'inconvénient majeur de ces approches provient alors de la non unicité des réseaux de régulation, en effet les réseaux de régulation ne sont pas statiques et évoluent dans le temps, des régulations peuvent ainsi se former ou disparaître en fonction de l'environnement. Rassembler les données de différentes expériences nécessitent donc de prendre en compte cet aspect dynamique de la structure.

Nous pouvons également nous interroger sur la pertinence des réseaux bayésiens pour ce problème biologique. Le caractère acyclique de ces réseaux semble jouer dès le départ en sa défaveur, rendant alors impossible la représentation de circuits d'auto régulation présents dans les réseaux biologiques. Cependant le caractère causal de certains arcs et l'aspect probabiliste du modèle justifie pleinement son utilisation dans ce cadre. De plus cette contrainte d'acyclicité peut être dépassée lors de l'utilisation de méthodes ensemblistes, où le résultat fourni n'est plus un unique réseau mais un classement de l'ensemble des relations possibles d'après une collection de réseaux appris. Ces approches constituent par ailleurs un choix pertinent lorsque le nombre d'observations est réduit, augmentant alors la robustesses des arcs prédits par rapport à ceux d'un unique réseau.

Les réseaux bayésiens sont donc pertinents pour ce type d'étude, cependant considérer uniquement cette méthode afin de reconstruire un réseau de régulation de gènes est peu réaliste. La généralisation d'approches effectuant le consensus de plusieurs méthodes semble alors plus judicieuse comme nous avons pu le montrer. L'apprentissage des réseaux bayésiens doit donc continuer à progresser, contribuant ainsi à l'amélioration de ce type de méta-analyse.

Malgré ce contexte difficile nous avons obtenu durant cette thèse des améliorations notables sur les deux axes de recherche. Tout d'abord sur l'apprentissage de la structure des réseaux bayésiens, un problème considéré depuis de nombreuses années et qui malgré tout reste encore ouvert à des améliorations algorithmiques. L'efficacité de nos propositions dans le cadre des recherches locales à base de score prouve en effet que les approches heuristiques peuvent encore être améliorées, se rapprochant ainsi des solutions optimales en terme de score. De même la qualité structurelle des réseaux appris peut être améliorée par une analyse locale de motifs récurrents comportant de fausses relations tel que nous l'avons montré avec la correction des *v-structures* couvertes.

Concernant le problème de reconstruction des réseaux de régulation de gènes, nous avons vu que l'utilisation de données combinées augmente la qualité de l'apprentissage. Nous avons également montré que dans le cas des données de *génétique-génomique*, des restrictions cohérentes avec les données de polymorphismes peuvent être appliquées au modèle, afin d'intégrer ces données à moindre coût en terme de complexité sans en détériorer la qualité. De plus, les réseaux bayésiens permettent dans ce cadre biologique d'attribuer un caractère causal à certaines relations entre gènes, venant ainsi préciser des situations observées par les analyses courantes en génétique telles que l'analyse eQTL. Cette causalité représente également un atout pour des approches consensus comme la méta-analyse. Nous notons justement que dans le cas d'approches consensus, les réseaux bayésiens permettent d'apprendre des régulations spécifiques, représentant par exemple des effets non linéaires, ce qui motive ainsi leur utilisation dans ce cas.

Les travaux présentés dans ce manuscrit ont fait l'objet de plusieurs publications ou présentations orales :

Publications :

« *Inférence de réseaux de régulation de gènes au travers de scores étendus dans les réseaux bayésiens* », Revue d'Intelligence Artificielle (RIA), vol. 26 :6, 2012.

« *Gene Regulatory Network Reconstruction Using Bayesian Networks, the Dantzig Selector, the Lasso and Their Meta-Analysis* », PLoS ONE, vol.6, 2011.
http://sites.google.com/site/vandeljimmy/Vignes12.pdf

Présentations orales :

« *New Local Move Operators for Bayesian Network Structure Learning* »
Workshop on Probabilistic Graphical Models, Spain, 2012.
http://sites.google.com/site/vandeljimmy/Vandel12c.pdf

« *New Local Move Operators for Learning the Structure of Bayesian Networks* »
ECAI'12 workshop, Algorithmic issues for inference in graphical models, Montpellier, 2012.
http://sites.google.com/site/vandeljimmy/Vandel12b.pdf

« *A New Local Move Operator for Reconstructing Gene Regulatory networks* »
CP'11 workshop, Constraint Based Methods for Bioinformatics, Italy, 2011.
http://sites.google.com/site/vandeljimmy/Vandel11a.pdf

« *Extended bayesian scores for reconstructing gene regulatory networks* »
ECCS'10 workshop, Graphical models for reasoning on biological systems, Portugal, 2010.
http://sites.google.com/site/vandeljimmy/Vandel10b.pdf

« *Reconstruction de réseau de régulation de gène à l'aide de données génomiques et de données génétiques* »
CAp'10, Conférence sur l'apprentissage automatique, Clermont-Ferrand, 2010.
http://sites.google.com/site/vandeljimmy/Vandel10a.pdf

Bibliographie

S. Acid and L. de Campos. A hybrid methodology for learning belief networks : Benedict. *International Journal of Approximate Reasoning*, 27(3) :235–262, 2001.

S. Acid and L. M. de Campos. Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs. *J. Artif. Int. Res.*, 18 :445–490, 2003.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19 :716–723, 1974.

C. Aliferis, I. Tsamardinos, and A. Statnikov. Large-scale feature selection using markov blanket induction for the prediction of protein-drug binding. Technical report, 2002.

C. Aliferis, I. Tsamardinos, and S. A. Hiton, a novel markov blanket algorithm for optimal variable selection. In *Proceedings of of AMIA*, pages 21–25, 2003a.

C. Aliferis, I. Tsamardinos, A. Statnikov, and L. Brown. Causal Explorer : A Probabilistic Network Learning Toolkit for Biomedical Discovery. In *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, pages 371–376, 2003b.

C. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i : Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11 :171–234, 2010.

J. Alonso-Barba, L. d. l. Ossa, and J. Puerta. Structural learning of bayesian networks using local algorithms based on the space of orderings. *Soft Comput*, pages 1881–1895, 2011.

J. Anderssona, R. Waltersb, P. Hortonb, and S. Janssona. Antisense Inhibition of the Photosynthetic Antenna Proteins CP29 and CP26 : Implications for the Mechanism of Protective Energy Dissipation. *Plant Cell*, 13 :1193–1204, 2001.

C. Auliac, V. Frouin, X. Gidrol, and F. d'Alché Buc. Evolutionary approaches for the reverse-engineering of gene regulatory networks : a study on a biologically realistic dataset. *BMC Bioinformatics*, 9 :91–104, 2008.

V. Auvray and L. Wehenkel. On the construction of the inclusion boundary neighbourhood for markov equivalence classes of bayesian network structures. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, UAI'02, pages 26–35. Morgan Kaufmann Publishers Inc., 2002.

G. P. Babu and M. N. Murty. A near-optimal initial seed value selection in k-means algorithm using a genetic algorithm. *Pattern Recogn. Lett.*, 14(10) :763–769, 1993.

O. Bangsø and P.-H. Wuillemin. Object oriented bayesian networks : A framework for topdown specification of large bayesian networks and repetitive structures. Technical report, Department of Computer Science, Aalborg University., Aalborg, Denmark, 2000.

A. Barabasi and Z. Oltvai. Network biology : Understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2) :101–115, 2004.

O. Barrière, E. Lutton, and P.-H. Wuillemin. Bayesian network structure learning using cooperative coevolution. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 755–762. ACM, 2009.

A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso : pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4) :791–806, 2011.

J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of Royal Statistical Society*, 36 :192–236, 1974.

R. Blanco, I. Inza, and P. Larrañaga. Learning bayesian networks in the space of structures by estimation of distribution algorithms. *International journal of intelligent systems*, 18 :205–220, 2003.

G. Bonaventure and J. Ohlrogge. Differential regulation of mRNA levels of acyl carrier protein isoforms in Arabidopsis. *Plant Physiology*, 128 :223–235, 2002.

R. Bouckaert. Probabilistic network construction using the minimum description length principle. In M. Clarke, R. Kruse, and S. Moral, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 747 of *Lecture Notes in Computer Science*, pages 41–48. Springer Berlin / Heidelberg, 1993.

L. Breiman, J. Friedman, R. Olsen, and C. Stone. *Classification and Regression Trees*. Wadsworth International, 1984.

L. d. Campos, J. Fernandez-Luna, and J. Puerta. Local Search Methods for Learning Bayesian Networks Using a Modified Neighborhood in the Space of DAGs. In *Advances in Artificial Intelligence — IBERAMIA 2002*, volume 2527, pages 182–192. 2002.

R. Castelo and A. Roverato. A robust procedure for gaussian graphical model search from microarray data with p larger than n. *Journal of machine learning research*, 7 :2621–2650, 2006.

G. Celeux and G. Govaert. A classification em algorithm for clustering and two stochastic versions. *Comput. Stat. Data Anal.*, 14(3) :315–332, 1992.

E. Chaibub Neto, C. Ferrara, A. Attie, and B. Yandell. Inferring causal phenotype networks from segregating populations. *Genetics*, 179(2), 6 2008.

J. Chen and Z. Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3) :759–711, 2008.

J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning bayesian networks from data : an information-theory based approach. *Artif. Intell.*, 137(1-2) :43–90, 2002.

H. Chernoff and E. Lehmann. The use of maximum likelihood estimates in x2 tests for goodness of fit. *The Annals of Mathematical Statistics*, 25 :579–586, 1954.

D. Chickering and D. Maxwell. Learning equivalence classes of bayesian-network structures. *J. Mach. Learn. Res.*, 2 :445–498, 2002.

D. Chickering, D. Heckerman, and C. Meek. Large-Sample Learning of Bayesian Networks is NP-Hard. *Journal of Machine Learning Research*, 5 :1287–1330, 2004.

K. Chipman and A. Singh. Using stochastic causal trees to augment bayesian networks for modeling eqtl datasets. *BMC Bioinformatics*, 12(1) :7–23, 2011.

J. Chiquet, A. Smith, G. Grasseau, C. Matias, and C. Ambroise. SIMoNe : Statistical Inference for MOdular NEtworks. *Bioinformatics*, 25(3) :417–418, 2009. doi : 10.1093/bioinformatics/btn637.

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3) :462–467, 1968.

J. Chu, S. Weiss, V. Carey, and B. Raby. A graphical model approach for inferring large-scale networks integrating gene expression and genetic polymorphism. *BMC Systems Biology*, 3(55), 2009.

G. Cooper and E. Hersovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9 :309–347, 1992.

F. Cubillos, J. Yansouni, H. Khalili, S. Balzergue, S. Elftieh, M. Martin-Magniette, Y. Serrand, L. Lepiniec, S. Baud, B. Dubreucq, J. Renou, C. Camilleri, and O. Loudet. Expression variation in connected recombinant populations of Arabidopsis thaliana highlights distinct transcriptome architectures. *BMC Genomics*, (13) :117–128, 2012.

C. de Campos, Z. Zeng, and Q. Ji. Structure learning of bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 113–120. ACM, 2009.

L. M. De Campos, J. M. Fernández-Luna, and J. M. Puerta. An iterated local search algorithm for learning bayesian networks with restarts based on conditional independence tests. *International Journal of Intelligent Systems*, 18(2) :221–235, 2003.

S. de Givry, M. Bouchez, P. Chabrier, D. Milan, and T. Schiex. CARTHAGENE : multipopulation integrated genetic and radiated hybrid mapping. *Bioinformatics*, 21(8) :1703–1704, 2005.

A. de la Fuente, N. Bing, I. Hoeschele, and P. Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18) :3565–3574, 2004.

T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Comput. Intell.*, 5 (3) :142–150, 1989.

A. Dobra, C. Hans, B. Jones, J. Nevins, G. Yao, and M. West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1) :196–212, 2004.

D. Edwards, G. de Abreu, and R. Labouriau. Selecting high-dimensional mixed graphical models using minimal aic or bic forests. *BMC Bioinformatics*, 11 :1–13, 2010.

G. Elidan, M. Ninio, N. Friedman, and D. Schuurmans. Data perturbation for escaping local maxima in learning. In *Proceedings of AAAI*, pages 132–139, 2002.

J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner. Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. *PLoS Biol*, 5(1) :54–66, 2007.

A. Favier, S. De Givry, and P. Jégou. Exploiting problem structure for solution counting. In *Proceedings of the 15th international conference on Principles and practice of constraint programming*, CP'09, pages 335–343, 2009.

P. Festa, P. M. Pardalos, and M. G. C. Resende. Feedback Set Problems. In *Encyclopedia of Optimization*, pages 1005–1016. Springer, 2009.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3) :432–441, 2008.

N. Friedman. Discretizing continuous attributes while learning bayesian networks. In *Proceedings of ICML*, pages 157–165. Morgan Kaufmann, 1996.

N. Friedman and D. Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, 50 :95–125, 2003.

N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with bayesian networks : a bootstrap approach. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, UAI'99, pages 196–205. Morgan Kaufmann Publishers Inc., 1999a.

N. Friedman, I. Nachman, and D. Peér. Learning bayesian network structure from massive datasets : Sparse candidate algorithm. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, UAI'99, pages 206–215. Morgan Kaufmann Publishers Inc., 1999b.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyse expression data. *Journal of computational biology*, 7(3/4) :601–620, 2000.

S. Gagnot, J.-P. Tamby, M.-L. Martin-Magniette, F. Bitton, L. Taconnat, S. Balzergue, S. Aubourg, J.-P. Renou, A. Lecharny, and V. Brunaud. CATdb : a public access to Arabidopsis transcriptome data from the URGV-CATMA platform. *Nucleic Acids Research*, 36(suppl 1) :D986–D990, 2008.

J. Gámez, J. Mateo, and J. Puerta. Learning Bayesian networks by hill climbing : efficient methods based on progressive restriction of the neighborhood. *Data Min. Knowl. Discov.*, 22 :106–148, 2011.

T. Gardner, D. di Bernardo, D. Lorenz, and J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629) :102–105, 2003.

S. B. Gillispie and M. D. Perlman. Enumerating markov equivalence classes of acyclic digraph dels. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, UAI'01, pages 171–177. Morgan Kaufmann Publishers Inc., 2001.

C. Giraud, S. Huet, and N. Verzelen. Graph selection with GGM select. Technical report, Ecole Polytechnique, 2009.

F. Glover and M. Laguna. Tabu search. *Modern Heuristic Techniques for Combinatorial Problems*, pages 70–150, 1993.

A. Goldenberg and A. Moore. Tractable learning of large Bayes net structures from sparse data. In *Proceedings of of ICML'04*, pages 44–51, 2004.

B. Haeupler, T. Kavitha, R. Mathew, S. Sen, and R. Tarjan. Faster algorithms for incremental topological ordering. In *Proceedings of of ICALP*, pages 421–433, 2008.

M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA Data Mining Software. *SIGKDD Explorations*, 11(1) :10–18, 2009.

C. Hans. Bayesian lasso regression. *Biometrika*, 96(4) :835–845, 2009.

A. Hartemink. *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks*. PhD thesis, Massachusetts Institute of Technology, 2001.

A. Hartemink. Reverse Engineering Gene Regulatory Networks. *Nature Biotechnology*, 23 :554–555, 2005.

A. Hauser and P. Bühlmann. Two optimal strategies for active learning of causal models from interventions. In *Proceedings of the Sixth edition of European Workshop on Probabilistic Graphical Models*, PGM'12, pages 123–130, 2012.

D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian Networks : The Combination of Knowledge and Statistical Data. In *Machine Learning*, volume 20, pages 197–243, 1995.

P. Hilson, J. Allemeersch, T. Altmann, S. Aubourg, A. Avon, J. Beynon, R. P. Bhalerao, F. Bitton, M. Caboche, B. Cannoot, V. Chardakov, C. Cognet-Holliger, V. Colot, M. Crowe, C. Darimont, S. Durinck, H. Eickhoff, A. F. de Longevialle, E. E. Farmer, M. Grant, M. T. Kuiper, H. Lehrach, C. Léon, A. Leyva, J. Lundeberg, C. Lurin, Y. Moreau, W. Nietfeld, J. Paz-Ares, P. Reymond, P. Rouzé, G. Sandberg, M. D. Segura, C. Serizet, A. Tabrett, L. Taconnat, V. Thareau, P. Van Hummelen, S. Vercruysse, M. Vuylsteke, M. Weingartner, P. J. Weisbeek, V. Wirta, F. R. Wittink, M. Zabeau, and I. Small. Versatile Gene-Specific Sequence Tags for Arabidopsis Functional Genomics : Transcript Profiling and Reverse Genetics Applications. *Genome Research*, 14(10b) :2176–2189, 2004.

A. Holland, M. Fathi, M. Abramovici, and M. Neubach. Competing fusion for bayesian applications. In *Proceedings of of IPMU 2008*, pages 378–385, 2008.

S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. COPASI–a COmplex PAthway SImulator. *Bioinformatics*, 22(24) :3067–3074, 2006.

V. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring regulatory networks from expression data using tree-based methods. *Plos ONE*, 5(9), 2010.

T. Jaakkola, D. Sontag, A. Globerson, and M. Meila. Learning Bayesian Network Structure using LP Relaxations. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 358–365, 2010.

A. K. Jain. Data clustering : 50 years beyond k-means. *Pattern Recogn. Lett.*, 31(8) :651–666, 2010.

R. Jansen and J. Nap. Genetical genomics : the added value from segregation. *Trends in genetics*, 17 (7) :388–391, July 2001.

M. Janžura and J. Nielsen. A simulated annealing-based method for learning bayesian networks from statistical data : Research articles. *Int. J. Intell. Syst.*, 21(3) :335–348, 2006.

F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer Publishing Company, Incorporated, 2nd edition, 2007.

R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–104. Plenum Press, 1972.

J. J. B. Keurentjes, J. Fu, I. R. Terpstra, J. M. Garcia, G. van den Ackerveken, L. B. Snoek, A. J. M. Peeters, D. Vreugdenhil, M. Koornneef, and R. C. Jansen. Regulatory network construction in arabidopsis by using genome-wide gene expression quantitative trait loci. *Proceedings of the National Academy of Sciences*, 104(5) :1708–1713, 2007.

C. Knoth, J. Ringler, J. Dangl, and T. Eulgem. Arabidopsis wrky70 is required for full rpp4-mediated disease resistance and basal defense against hyaloperonospora parasitica. *Molecular Plant-Microbe Interactions*, 20 :120–128, 2007.

D. Koller and N. Friedman. *Probabilistic Graphical Models : Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

D. Koller and M. Sahami. Toward optimal feature selection. Technical Report 1996-77, Stanford InfoLab, 1996.

P. Kontkanen and P. Myllymäki. A linear-time algorithm for computing the multinomial stochastic complexity. *Information Processing Letters*, 103(6) :227 – 233, 2007.

W. Lam and F. Bacchus. Using causal information and local measures to learn bayesian networks. In *Proceedings of the Ninth international conference on Uncertainty in artificial intelligence*, UAI'93, pages 243–250. Morgan Kaufmann Publishers Inc., 1993.

P. Larrañaga, M. Poza, Y. Yurramendi, R. Murga, and C. Kuijpers. Structure learning of bayesian networks by genetic algorithms : A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 :912–926, 1996.

P. Leray, S. Meganek, S. Maes, and B. Manderick. Causal graphical models with latent variables : Learning and inference. *Innovations in Bayesian Networks*, 156 :219–249, 2008.

B. Liu, A. de la Fuente, and I. Hoeschele. Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics*, 178(3) :1763–1776, 2008.

B. Logsdon, G. Hoffman, and J. Mezey. Mouse obesity network reconstruction with a variational bayes algorithm to employ aggressive false positive control. *BMC Bioinformatics*, 13 :1–15, 2012.

B. A. Logsdon and J. Mezey. Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations. *PLoS Comput Biol*, 6(12), 12 2010.

J. L. Lustgarten, S. Visweswaran, V. Gopalakrishnan, and G. F. Cooper. Application of an efficient bayesian discretization method to biomedical data. *BMC Bioinformatics*, 12(1) :309, 2011.

D. Madigan and J. York. Bayesian graphical models for discrete data. *International Statistical Review*, 63 :215–232, 1995.

A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Favera, and A. Califano. Aracne : An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(Suppl 1), 2006.

S. Meganck, P. Leray, and B. Manderick. Learning causal bayesian networks from observations and experiments : A decision theoretic approach. *Modeling Decisions for Artificial Intelligence*, 3885 : 58–69, 2006.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The annals of statistics*, 34(3) :1436–1462, 2006.

P. Mendes, W. Sha, and K. Ye. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19 :122–129, 2003.

L. Michel and P. V. Hentenryck. A Constraint-Based Architecture for Local Search. In *Proceedings of 17th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pages 83–100, 2002.

S. Monti and G. F. Cooper. A multivariate discretization method for learning bayesian networks from mixed data. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, pages 404–413. Morgan Kaufmann Publishers Inc., 1998.

A. Moore and W. Wong. Optimal reinsertion : A new search operator for accelerated and more accurate bayesian network structure learning. In *Proceedings of of ICML '03*, pages 552–559, 2003.

K. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics : Proceedings of the Interface*, 33 :331–351, 2001.

P. Naim, P.-H. Wuillemin, P. Leray, and O. Pourret. *Réseaux bayésiens*. Eyrolles, 3 edition, 11 2007.

H.-T. Nguyen. *Réseaux bayésiens et apprentissage ensembliste pour l'étude différentielle de réseaux de régulation génétique*. PhD thesis, Université de Nantes, 2012.

J. Nielsen, T. Kocka, and J. Pefia. On Local Optima in Learning Bayesian Networks. In *Proceedings of of UAI-03*, pages 435–442, 2003.

A. Nikitin, S. Egorov, N. Daraselia, and I. Mazo. Pathway studio - the analysis and navigation of molecular networks. *Bioinformatics*, 19(16), 2003.

R. Pal, S. Bhattacharya, and M. Caglar. Robust approaches for genetic regulatory network modeling and intervention : A review of recent advances. *Signal Processing Magazine*, 29(1) :66–76, 2012.

J. Pearl. Bayesian networks : a model of self-activated memory for evidential reasoning. Technical report, Computer Science Department, University of California, 1985.

J. Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.

J. Pearl and T. S. Verma. A Theory of Inferred Causation. In J. F. Allen, R. Fikes, and E. Sandewall, editors, *KR'91 : Principles of Knowledge Representation and Reasoning*, pages 441–452, San Mateo, California, 1991. Morgan Kaufmann.

A. Pinna, N. Soranzo, I. Hoeschele, and A. de la Fuente. Simulating systems genetics data with sysgensim. *Bioinformatics*, 27(17) :2459–2462, 2011.

I. Pournara and L. Wernisch. Reconstruction of gene networks using bayesian learning and manipulation experiments. *Bioinformatics*, 20(17) :2934–2942, 2004.

R. Ram and M. Chetty. A guided genetic algorithm for learning gene regulatory networks. In *IEEE Congress on Evolutionary Computation'07*, pages 3862–3869, 2007.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14 :465–471, 1978.

R. Robinson. Counting unlabeled acyclic digraphs. *In Combinatorial Mathematics V*, 622 :28–43, 1977.

M. Rockman. Reverse engineering the genotype–phenotype map with natural genetic variation. *Nature*, 456, 12 2008.

E. E. Schadt, J. Lamb, X. Yang, J. Zhu, S. Edwards, D. Guhathakurta, S. K. Sieberts, S. Monks, M. Reitman, C. Zhang, P. Y. Y. Lum, A. Leonardson, R. Thieringer, J. M. Metzger, L. Yang, J. Castle, H. Zhu, S. F. Kash, T. A. Drake, A. Sachs, and A. J. Lusis. An integrative genomics approach to infer causal associations between gene expression and disease. *Nature genetics*, 37(7) : 710–717, 2005.

J. Schäfer and K. Strimmer. An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6) :754–764, 2005.

R. Scheines, P. Spirtes, C. Glymour, C. Meek, and T. Richardson. The TETRAD Project : Constraint Based Aids to Causal Model Specification. *Multivariate Behavioral Research*, 33(1) :65–117, 1998.

M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using L1-regularization paths. In *Proceedings of of AAAI'07*, pages 1278–1283, 2007.

G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 1978.

J. Scott and C. Carvalho. Feature-inclusion stochastic search for gaussian graphical models. *Journal of Computational and Graphical Statistics*, 17(4) :790–808, 2008.

M. Scutari. Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software*, 35 :1–22, 2010.

C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 : 379–423, 1948.

Y. M. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23(3) :175–186, 1987.

T. Silander, P. Kontkanen, and P. Myllymäki. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In *Proceedings of of UAI-07*, pages 360–367, Vancouver, Canada, 2007.

T. Silander, T. Roos, and P. Myllymäki. Learning locally minimax optimal bayesian networks. *International Journal of Approximate Reasoning*, 51(5) :544 – 557, 2010.

C. Sima, J. Hua, and S. Jung. Inference of gene regulatory networks using time-series data : A survey. *Current Genomics*, 10(6) :416–429, 2009.

M. Simon, O. Loudet, S. Durand, A. Bérard, D. Brunel, F.-X. Sennesal, M. Durand-Tardif, G. Pelletier, and C. Camilleri. QTL mapping in five new large RIL populations of Arabidopsis thaliana genotyped with consensus SNP markers. *Genetics*, 178 :2253–2264, 2008.

P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1) :62–72, 1991.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer-Verlag, 1993.

G. Stolovitzky, D. Monroe, and A. Califano. Dialogue on reverse-engineering assessment and methods. *Annals of the New York Academy of Sciences*, 1115(1) :1–22, 2007.

J. Suzuki. Learning bayesian belief networks based on the minimum description length principle : Basic properties. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 82(10) :2237–2245, 1999.

D. Swarbreck, C. Wilks, P. Lamesch, T. Berardini, M. Garcia-Hernandez, H. Foerster, D. Li, T. Meyer, R. Muller, L. Ploetz, A. Radenbaugh, S. Singh, V. Swing, C. Tissier, P. Zhang, and E. Huala. The Arabidopsis Information Resource (TAIR) : gene structure and function annotation. *Nucleic Acids Research*, 36 :1009–1014, 2008.

Y. Tamada, S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara, and S. Miyano. Estimating gene networks from gene expression data by combining bayesian network model with promoter element detection. *Bioinformatics*, 19(suppl 2), 2003.

M. Tan, M. Alshalalfa, R. Alhajj, and F. Polat. Influence of prior knowledge in constraint-based learning of gene regulatory networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 8(1) :130–142, 2011.

M. Teyssier and D. Koller. Ordering-based Search : A Simple and Effective Algorithm for Learning Bayesian Networks. In *Proceedings of of UAI'05*, pages 584–590, 2005.

G. Thibault, A. Aussem, and S. Bonnevay. Analyse critique des algorithmes EDA dans le cadre de l'apprentissage de structure de réseaux Bayésiens. In *Journées Francophone sur les Réseaux Bayésiens*, Lyon, France, 2008. 12 pages.

S. Tong and D. Koller. Active learning for structure in bayesian networks. In *in international joint conference on artificial intelligence*, pages 863–869, 2001.

I. Tsamardinos, C. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. In *Proceedings of the 16th International FLAIRS Conference*, pages 376–380. AAAI Press, 2003a.

I. Tsamardinos, C. Aliferis, and A. Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of of KDD'03*, pages 673–678, 2003b.

I. Tsamardinos, L. Brown, and C. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Mach. Learn.*, 65 :31–78, 2006.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings Sixth Conference on Uncertainty and Artificial Intelligence*, pages 255–270, San Francisco, 1990. Morgan Kaufmann.

N. Verzelen. High-dimensional gaussian model selection on a gaussian design. *Annales de l' institut H. Poincaré Probabilités et Statistiques*, 46(2) :480–524, 2010.

M. Vignes, J. Vandel, D. Allouche, N. Ramadan-Alban, C. Cierco-Ayrolles, T. Schiex, B. Mangin, and S. de Givry. Gene Regulatory Network Reconstruction Using Bayesian Networks, the Dantzig Selector, the Lasso and Their Meta-Analysis. *PLoS ONE*, 6, 2011.

F. Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1 :80–83, 1945.

J. Wildenhain and E. Crampin. Reconstructing gene regulatory networks : from random to scale-free connectivity. *Systems Biology, IEE Proceedings*, 153(4) :247–256, 2006.

A. Wille and P. Bühlmann. Low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, 5(1), 2006.

M. L. Wong and K. S. Leung. An efficient data mining method for learning bayesian networks using an evolutionary algorithm-based hybrid approach. *Trans. Evol. Comp*, 8(4) :378–404, 2004.

M. L. Wong, W. Lam, and K. S. Leung. Using evolutionary programming and minimum description length principle for data mining of bayesian networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21 (2) :174–178, 1999.

J. Yu, V. Smith, P. Wang, A. Hartemink, and E. Jarvis. Using bayesian network inference algorithms to recover molecular genetic regulatory networks. In *Proceedings of International Conference on Systems Biology*, 2002.

C. Yuan, B. Malone, and X. Wu. Learning optimal Bayesian networks using A* search. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, 2011.

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94 (1) :19–35, 2007.

D. Zhu and H. Li. Improved bayesian network inference using relaxed gene ordering. *Int. J. Data Min. Bioinformatics*, 4(1) :44–59, 2010.

J. Zhu, M. C. Wiener, C. Zhang, A. Fridman, E. Minch, P. Y. Lum, J. R. Sachs, and E. E. Schadt. Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. *PLOS COMPUTATIONAL BIOLOGY*, 3(4) :692–703, 2007.

J. Zhu, B. Zhang, E. N. Smith, B. Drees, R. B. Brem, L. Kruglyak, R. E. Bumgarner, and E. E. Schadt. Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature Genetics*, 40(7) :854–861, 2008.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476) :1418–1429, 2006.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(2) :301–320, 2005.


SIGILLVM·SCOLARIVM·TOLOSE·
VNIVERSITATIS·MAGISTRORVM·ET·SCOLARIVM

Résumé

Apprendre la structure d'un réseau de régulation de gènes est une tâche complexe due à la fois au nombre élevé de variables le composant (plusieurs milliers) et à la faible quantité d'échantillons disponibles (quelques centaines). Parmi les approches proposées, nous utilisons le formalisme des réseaux bayésiens, ainsi apprendre la structure d'un réseau de régulation consiste à apprendre la structure d'un réseau bayésien où chaque variable représente un gène et chaque arc un phénomène de régulation. Dans la première partie de ce manuscrit nous nous intéressons à l'apprentissage de la structure de réseaux bayésiens génériques au travers de recherches locales. Nous explorons plus efficacement l'espace des réseaux possibles grâce à un nouvel algorithme de recherche stochastique (SGS), un nouvel opérateur local (SWAP), ainsi qu'une extension des opérateurs classiques qui permet d'assouplir temporairement la contrainte d'acyclicité des réseaux bayésiens. La deuxième partie se focalise sur l'apprentissage de réseaux de régulation de gènes. Nous proposons une modélisation du problème dans le cadre des réseaux bayésiens prenant en compte deux types d'information. Le premier, classiquement utilisé, est le niveau d'expression des gènes. Le second, plus original, est la présence de mutations sur la séquence d'ADN pouvant expliquer des variations d'expression. L'utilisation de ces données combinées dites de génétique-génomique, vise à améliorer la reconstruction. Nos différentes propositions se sont montrées performantes sur des données de génétique-génomique simulées et ont permis de reconstruire un réseau de régulation à partir de données observées sur le plante Arabidopsis thaliana.

Abstract

Structure learning of gene regulatory networks is a complex process, due to the high number of variables (several thousands) and the small number of available samples (few hundred). Among the proposed approaches to learn these networks, we use the Bayesian network framework. In this way to learn a regulatory network corresponds to learn the structure of a Bayesian network where each variable is a gene and each edge represents a regulation between genes. In the first part of this thesis, we are interested in learning the structure of generic Bayesian networks using local search. We explore more efficiently the search space thanks to a new stochastic search algorithm (SGS), a new local operator (SWAP) and an extension for classical operators to briefly overcome the acyclic constraint imposed by Bayesian networks. The second part focuses on learning gene regulatory networks. We proposed a model in the Bayesian networks framework taking into account two kinds of information. The first one, commonly used, is gene expression levels. The second one, more original, is the mutations on the DNA sequence which can explain gene expression variations. The use of these combined data, called genetical genomics, aims to improve the structural learning quality. Our different proposals appeared to be efficient on simulated genetical genomics data and allowed to learn a regulatory network from observed data from Arabidopsis thaliana.