# Estimation de la structure d'indépendance conditionnelle d'un réseau de capteurs : application à l'imagerie médicale

Aude Costard

UNIVERSITÉ DE GRENOBLE

**THÈSE**

Pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE**

Spécialité : **Signal, Image, Parole, Télécoms (SIPT)**

Arrêté ministériel : 7 août 2006

Présentée par

**Aude COSTARD**

Thèse dirigée par **Olivier MICHEL** et **Patrice ABRY**
co-encadrée par **Sophie ACHARD** et **Pierre BORGNAT**

préparée au sein du **Laboratoire Grenoble Images Parole Signal et Automatique (GIPSA-lab)** et du **Laboratoire de Physique de l'École Normale Supérieure (ENS) de Lyon**
dans **L'École Doctorale d'Électronique, d'Électrotechnique, d'Automatique et du Traitement du Signal (EEATS)**

# Estimation de la structure d'indépendance conditionnelle d'un réseau de capteurs. Application à l'imagerie médicale.

Thèse soutenue publiquement le **10 novembre 2014**,
devant le jury composé de :

**Michel DESVIGNES**
Professeur, Grenoble-INP, Président du jury
**Christian HEINRICH**
Professeur, Université de Strasbourg, Rapporteur
**Olivier LÉZORAY**
Professeur, Université de Caen, Rapporteur
**Fabrizio DE VICO FALLANI**
Chercheur, INRIA, Examinateur
**Djalel MESKALDJI**
Research Associate, EPFL, Examinateur
**Olivier MICHEL**
Professeur, Grenoble-INP, Directeur de thèse
**Sophie ACHARD**
CR, CNRS - Université de Grenoble, Encadrante de thèse
**Pierre BORGNAT**
CR, CNRS - ENS Lyon, Encadrant de thèse




*Université Joseph Fourier / Université Pierre Mendès France / Université Stendhal / Université de Savoie / Grenoble INP*

# Remerciements

Au moment où j'écris ces lignes, j'en suis arrivée à la conclusion que la thèse c'est comme une campagne menée à bord d'un magnifique galion de l'armée de la science. On monte à bord plein d'ambition et de suffisance, en rêvant de découvertes plus époustouflantes les unes que les autres, découvertes qui nous mènerons vers des terres inconnues. Pour moi ça a commencé sur une mer calme peuplée de petites iles désertes, chacune différente, parfois découvertes par hasard car non présentes sur mes cartes. Puis sont apparus les premiers écueils et le brouillard, suivis des tempêtes. Puis la mer se calme, on se retrouve au milieu de l'océan avec uniquement de l'eau aussi loin que porte le regard. Puis un beau jour ce mot qui résonne "TERRE". La terre promise est en vue. C'est avec joie et appréhension qu'on débarque sur cette terre nouvelle pleine de promesses. Dans les quelques lignes qui suivent, je tiens à remercier toutes les personnes qui ont participé à l'aventure.

Je voudrais en tout premier remercier mes parents qui m'ont regardé monter à bord, aussi ignorants que moi de ce qui m'attendait mais en me donnant leur bénédiction. Ils m'ont soutenue tout au long du voyage, un peu désemparés devant l'ampleur de la tâche mais toujours avec beaucoup d'amour. Merci du fond du cœur.

Je tiens également à remercier mes capitaines, Sophie A., Olivier M., Pierre B. et Patrice A. notamment pour m'avoir fait embarquer à bord de ce navire, pour m'avoir donner mes premières cartes d'exploration, pour avoir fait face avec moi aux tempêtes et pour avoir essuyé les tentatives de mutinerie. Finalement nous sommes arrivés à bon port ! Et vous avez fait du petit mousse un marin plus expérimenté, prêt pour de nouvelles aventures (mais pas plus grand pour autant !).

Merci à mon comité d'accueil en terre promise, Michel D., Christian H., Olivier L., Fabrizio D.V.F. et Djalel M. pour l'intérêt qu'ils ont manifesté pour le fruit de ces trois années de campagne ainsi que pour leurs remarques constructives.

Merci à la région Rhône-Alpes d'avoir financer mon expédition.

Un grand merci à l'ensemble de la flotte du GIPSA-lab, pour la bonne humeur qui règne à bord et la disponibilité de chacun.

Un merci plus particulier à mes compagnons de galères. Merci à l'équipage déjà à bord ou ayant embarqué en même temps que moi, Cyrille, Flore, Gailene, Jérémy, Jonathan, Luc, Matthieu, Vincent et Wei, pour nos folles parties. Wei, xièxie pour avoir été une élève modèle, curieuse et enthousiaste, ça a été un plaisir de m'improviser prof de français. Merci à la livraison de nouveaux matelots arrivés à bord au bout d'un an (ou un peu plus pour certain), Arnaud, Céline, Cindy, Edouard, Manu, Pascal, Quentin, Raph et Tim, pour les gâteaux, les discussions à vous mettre le cerveau à l'envers, les tueries de zombies, la queue devant les micro-ondes, pour m'avoir fait passer pour une mamie et j'en passe. Merci à la dernière génération, Alexis, Lucas, Taïa, pour avoir apporté un peu de sang frais car la deuxième génération commençait à

se transformer en papis.

Merci aux différents bureaux du gipsa-doc pour avoir animé avec brio la vie des matelots du GIPSA. Un merci particulier à Humberto, Ignacio, Jack, Soheib et Yo pour avoir tenu les rênes avec moi pendant un an.

Merci à Fanny grâce à qui mes travaux ont pu être montrés sous leur meilleur jour mais également pour son enthousiasme. Je te souhaite le meilleur pour la suite.

Merci à tous les autres qui ont fait partie intégrante de ma vie à bord et que je tiens à remercier pour les discussions d'ordre scientifique, psychologique voire gastronomique que nous avons partagé : Antoine, Carole, Dana, Delphine, Fakhri, Fatima, Florian, Francesca, Gildas, Guillaume, Hanna, Hélène, Jérémy, Ladan, Nikola, Maëlle, Maël, Marc, Olivier, Raluca, Rémy, Rodrigo, Stefen, Tim et plus encore.

Merci aux matelots des autres flottes que j'ai côtoyés au sein des différents conseils (de l'ED EEATS et du collège doctoral) pour m'avoir permis de m'ouvrir à d'autres horizons. Un merci spécial à Emmanuelle avec qui j'ai refait le monde plus d'une fois. J'espère qu'on aura apporté notre grain de sable à l'édifice.

Merci également à Nicolas et Ronan qui m'ont toujours accueilli chaleureusement lors de mes passages à bord du navire de l'ENS.

Merci maintenant à ceux qui ne liront surement jamais ces lignes, à ceux qui m'ont permis de m'échapper de mon quotidien de matelot : les Argonautes et les membres du KCG. Merci également à la troupe de Carnage (je ne savais pas où vous mettre, quelle idée de faire du théâtre alors qu'on est en thèse !).

Je remercie maintenant tous ceux que j'ai oublié de remercier, la famille, la belle-famille, les H20, les potes de prépas, les cOpines, les potes de SICOM, et ceux qui ne rentrent dans aucune de ces cases.

Et j'ai bien sûr gardé le meilleur pour la fin, merci à mon co-rameur, merci pour tout. Je t'attend en terre promise pour qu'on continue l'aventure ensemble.

Je finirai sur ces mots : A ceux qui sont encore à bord "BON VENT" et à ceux qui hésitent à monter "A L'ABORDAGE".

# Table des matières

# Notations

| | |
|---|---|
| $A$ | matrice d'adjacence |
| $C$ | clique |
| $E$ | ensemble des arêtes d'un graphe $G$ |
| $G$ | graphe |
| $K$ | matrice de précision |
| $\mathbb{P}(X)$ | densité de probabilité de la variable $X$ |
| $\mathbb{P}_X(x)$ | probabilité que la variable $X$ vaille $x$ |
| $S$ | séparateur |
| $\mathbf{X}$ | processus multivarié |
| $X_i$ | $i^{\text{ème}}$ variable du processus multivarié $\mathbf{X}$ |
| $V$ | ensemble des nœuds d'un graphe $G$ |
| $V(G)$ | volume des matrices de corrélations partielles ayant des zéros là où $G$ n'a pas d'arêtes |
| $W$ | matrice d'Isserlis |
| $d$ | degré d'un nœud |
| $e$ | paramètre de voisinage |
| $n$ | nombre d'observations |
| $p$ | nombre de variables pour un processus multivarié ET nombre de nœuds pour un graphe $G$ |
| $s$ | seuil pour générer les matrices des corrélations partielles parcimonieuses |
| $\mathcal{C}$ | ensemble des cliques d'un graphe |
| $\mathcal{H}_0$ et $\mathcal{H}_1$ | pour un test d'hypothèse, hypothèse nulle et hypothèse alternative |
| $\mathcal{IW}_p(\delta, U)$ | loi de Wishart inverse de degré de liberté $\delta$ et de matrice d'échelle $U$ |
| $\mathcal{N}(\mu, \Sigma)$ | loi gaussienne multivariée de moyenne $\mu$ et de covariance $\Sigma$ |
| $\mathcal{S}$ | ensemble des séparateurs d'un graphe |
| $\mathcal{W}_p(\delta, U)$ | loi de Wishart de degré de liberté $\delta$ et de matrice d'échelle $U$ |
| $\Pi$ | matrice des corrélations partielles (sa version vectorisée est notée $\boldsymbol{\pi}$) |
| $\Sigma$ | matrice de covariance |
| $\Sigma^{emp}$ | matrice de covariance empirique |
| $\Phi_{\mu,\Sigma}$ | loi de répartition de la loi gaussienne de moyenne $\mu$ et de covariance $\Sigma$ |
| $\alpha$ | risque de première espèce (ou risque de faux positif) |
| $\beta_{ij}$ | coefficient de régression linéaire entre les variables $X_i$ et $X_j$ |
| $\gamma$ | support d'un graphe $G$ |
| $\epsilon_i$ | résidu de $X_i$ sachant $X_{V\setminus\{i\}}$ |
| $\epsilon_{ij}$ | résidu de $X_i$ sachant $X_{V\setminus\{i,j\}}$ |
| $\lambda$ | paramètre de pénalisation |
| $\boldsymbol{\pi}$ | vecteur des corrélations partielles selon le support du graphe $G$ (sa version matricielle est notée $\Pi$) |
| $\varphi_{\mu,\Sigma}(\mathbf{y})$ | fonction gaussienne de moyenne $\mu$ et de covariance $\Sigma$ appliquée au vecteur $\mathbf{y}$ |
| $\perp\!\!\!\perp$ | symbole d'indépendance |
| $\cdot \perp\!\!\!\perp \cdot \vert \cdot$ | symbole d'indépendance conditionnelle |
| $\vert \cdot \vert$ | déterminant OU cardinal d'un ensemble |

# Introduction

Le présent manuscrit présente les travaux effectués dans le cadre de ma thèse de doctorat. Ces travaux s'inscrivent dans le contexte de l'étude de réseaux de capteurs.

## Capteur

***Capteur*** *n.m. : Organe qui élabore, à partir d'une grandeur physique, une autre grandeur physique, souvent de nature électrique, utilisable à des fins de mesure ou de commande.*

Dictionnaire Larousse

Les capteurs peuvent mesurer des grandeurs qui évoluent au cours du temps comme la température ou des événements plus ponctuels comme l'apparition de fumée. La figure 1 donne des exemples de capteurs de la vie courante.


(a) Capteur de température

(b) Capteur de fumée


(c) Bouton d'arrêt d'urgence

FIGURE *1 – Différents exemples de capteurs.*

En plus des capteurs traditionnels (construits pour mesurer une grandeur physique), nous considérons comme capteur tout élément auquel il est possible d'associer un signal qui est la mesure d'un phénomène physique. Par exemple, en imagerie cérébrale, il est possible d'enregistrer des images de la totalité du cerveau à différents instants. Si on associe à une région du cerveau l'évolution de sa valeur dans l'image entre les différents instants d'acquisition, la région considérée peut être assimilée à un capteur.

## Réseaux de capteurs

Un réseau de capteurs est un ensemble de capteurs qui interagissent entre eux. La figure 2 donne deux exemples de réseaux de capteurs.

Les interactions entre capteurs ont deux origines possibles :

– soit les capteurs communiquent entre eux, c'est-à-dire qu'ils s'envoient de l'information. C'est le cas d'un réseau de téléphonie mobile : le signal est transmis entre différents émetteurs-récepteurs (capteurs) afin de rendre possible la communication entre deux utilisateurs.

(a) Casque d'électro-encéphalographie

(b) Station de base (téléphonie mobile)

FIGURE *2 – Exemples de réseaux de capteurs*

– soit les capteurs enregistrent des phénomènes physiques sans échanger d'information. L'interaction provient du fait qu'un même phénomène peut être enregistré par plusieurs capteurs, au même instant ou à des instants différents.

Nous nous concentrons sur l'étude de réseaux de capteurs pour lesquels les interactions sont engendrées par les phénomènes physiques, les capteurs considérés ne communiquent pas directement entre eux.

Lorsque les interactions entre capteurs sont dues aux phénomènes physiques, deux types d'interactions sont observables :

– les interactions dirigées : un phénomène est enregistré par un capteur à un instant $t$ et par un deuxième capteur à un instant $t + \tau$.
– les interactions non dirigées : deux capteurs enregistrent le même phénomène au même instant.

La figure 3 illustre le principe d'une interaction dirigée entre deux capteurs. L'événement arrivant au temps $t_1$ est enregistré par un des capteurs au temps $t_2$ puis par l'autre capteur au temps $t_4$. Dans cet exemple, l'interaction est dirigée de $c_1$ vers $c_2$.

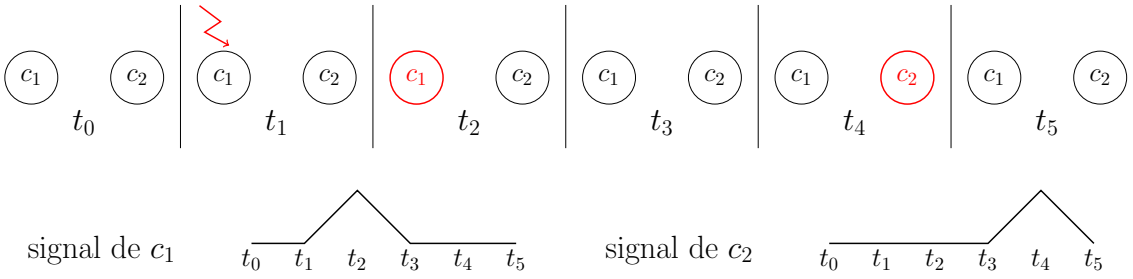



FIGURE *3 – Schéma simplifié d'une interaction dirigée entre deux capteurs. Un événement apparaît au temps $t_1$, le capteur $c_1$ l'enregistre au temps $t_2$ et le capteur $c_2$ l'enregistre au temps $t_4$.*

La figure 4 illustre le principe d'une interaction non dirigée entre deux capteurs. Les capteurs enregistrent le même événement au même instant. Ici le premier événement au temps $t_0$ (en vert) n'est enregistré que par le capteur $c_1$ au temps $t_1$ mais le second événement au temps $t_3$ (en bleu) est enregistré par les deux capteurs au temps $t_4$.

Nous nous intéressons uniquement à des réseaux de capteurs dont les interactions sont non dirigées car nous nous focalisons sur l'étude de réseaux de capteurs pour lesquels le temps entre deux acquisitions est grand devant la durée de propagation des phénomènes physiques mesurés.

Prenons un exemple simple pour comprendre l'influence du temps d'intégration sur le type

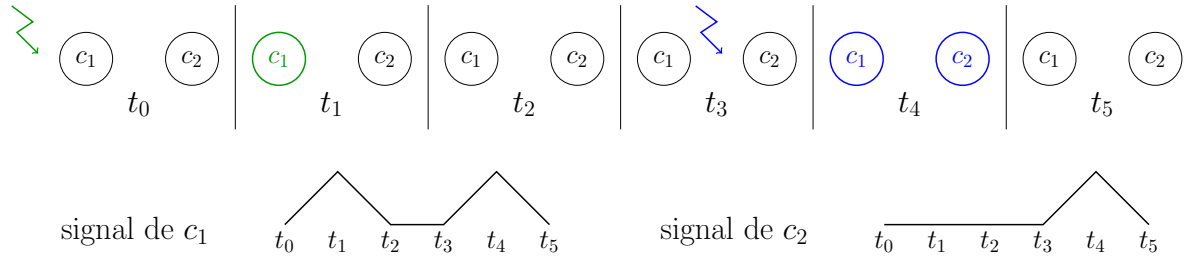



FIGURE 4 – *Schéma simplifié d'une interaction non dirigée entre deux capteurs. L'événement qui apparaît au temps $t_0$ n'est enregistré que par le capteur $c_1$ au temps $t_1$. Par contre l'événement qui apparaît au temps $t_3$ est enregistré par les deux capteurs aux temps $t_4$.*

d'interaction en deux capteurs. Considérons deux capteurs acoustiques distants de cinq kilomètres. La foudre tombe à 1 kilomètre du premier capteur et 6 kilomètres du deuxième. En approximant que le son se déplace à une vitesse de 340 m/s, si les capteurs ont un temps d'intégration de l'ordre de la seconde, le premier capteur donne l'information du coup de tonnerre environ 3 secondes après que la foudre soit tombée et le deuxième capteur donne l'information du coup de tonnerre environ 18 secondes après que la foudre soit tombée. L'interaction entre les capteurs est dirigée du premier capteur vers le deuxième. Si le temps d'intégration des capteurs est de l'ordre de la minute, les deux capteurs fournissent l'information du coup de tonnerre au même moment et l'interaction est alors perçue comme non-dirigée.

### Structure d'un réseau de capteurs

Dans le cadre de l'étude de réseaux de capteurs, il est important de connaître comment interagissent les capteurs, c'est-à-dire de savoir si deux capteurs enregistrent ou non les mêmes phénomènes physiques et dans quelle proportion. Reprenons l'exemple de la figure 4, les capteurs $c_1$ et $c_2$ enregistrent tous les deux le phénomène bleu mais le phénomène vert est enregistré uniquement pas le capteur $c_1$. Cela signifie qu'il existe une dépendance entre les capteurs $c_1$ et $c_2$ mais que cette dépendance n'est pas totale dans le sens où l'information enregistrée par un des deux capteurs n'est pas obligatoirement enregistrée par le deuxième capteur. Connaître les relations de dépendance entre tous les capteurs d'un réseau de capteurs revient à connaître sa structure de dépendance. La figure 5 donne un exemple de structure de dépendance pour un réseau de six capteurs. La "force" de la dépendance est proportionnelle à l'épaisseur des traits reliant les capteurs. Le couple de capteurs ($c_1$,$c_2$) est très dépendant, les couples ($c_1$,$c_3$), ($c_2$,$c_6$) et ($c_3$,$c_6$) sont très faiblement dépendants et les couples ($c_5$,$c_6$), ($c_3$,$c_5$) et ($c_3$,$c_4$) sont entre les deux et ordonnés par dépendance décroissante. Les couples n'étant pas reliés par des traits sont considérés comme indépendants.

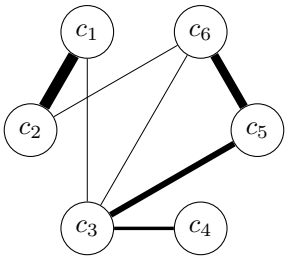



FIGURE 5 – *Exemple de structure de dépendance pour un réseau de 6 capteurs. La "force" de la dépendance est représentée par l'épaisseur des traits entre les capteurs, plus le trait est épais, plus la dépendance est importante.*

La notion de dépendance n'a pas de définition précise dans le cas général. De plus, l'estimation de la "force" de dépendance entre deux capteurs pouvant être problématique, l'étude de la structure de dépendance peut être simplifiée en se concentrant sur l'étude de la structure

d'indépendance : soit deux capteurs dépendent l'un de l'autre, soit il n'existe aucune dépendance entre les deux capteurs et ils sont considérés comme indépendants. La figure 6 donne la structure d'indépendance du réseau de six capteurs dont la structure de dépendance est donnée à la figure 5. Les traits sont tous de la même épaisseur puisque la relation est binaire : soit les capteurs sont dépendants et il y a un trait entre les capteurs, soit ils sont indépendants et il n'y a pas de trait.

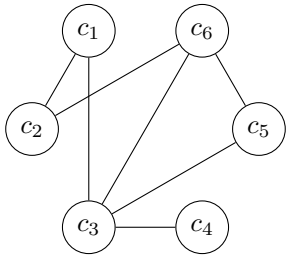
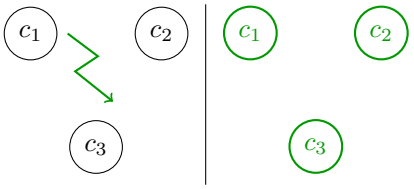



FIGURE 6 – *Exemple de structure d'indépendance pour le réseau de 6 capteurs dont la structure de dépendance est donnée figure 5. Les traits sont tous de la même épaisseur puisque la relation est binaire : soit les capteurs sont dépendants et il y a un trait entre les capteurs, soit ils sont indépendants et il n'y a pas de trait.*

Une approche complémentaire à celle de l'étude de la structure de dépendance est celle de l'étude de la structure de dépendance conditionnelle. La dépendance conditionnelle entre deux capteurs correspond à la dépendance entre ces deux capteurs connaissant l'information enregistrée par les autres capteurs du réseau. Prenons un exemple simple d'un réseau à trois capteurs. La figure 7 illustre une configuration où les trois capteurs enregistrent le même événement. Sachant l'information enregistrée par le capteur $c_3$, l'information restante dans les capteurs $c_1$ et $c_2$ n'est que du bruit et donc les deux capteurs sont indépendants conditionnellement à $c_3$. La figure 8 illustre une configuration où les capteurs $c_1$ et $c_2$ enregistrent deux événements distincts que nous supposons indépendants, les deux capteurs sont donc indépendants. Cependant le capteur $c_3$ enregistre les deux événements et donc connaissant l'information enregistrée par $c_3$, les deux capteurs $c_1$ et $c_2$ sont dépendants conditionnellement à $c_3$.



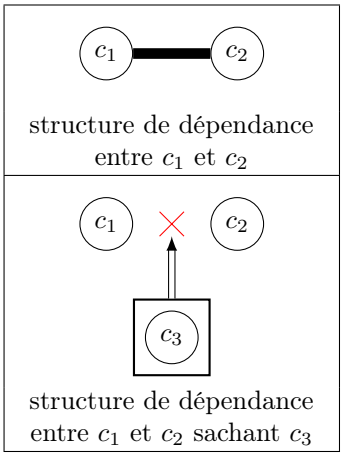




FIGURE 7 – *Exemple d'un réseau de 3 capteurs qui enregistrent simultanément le même événement. En haut à droite, la structure de dépendance entre les capteurs $c_1$ et $c_2$ d'après le comportement de l'exemple et en bas à droite, la structure de dépendance entre les capteurs $c_1$ et $c_2$ sachant l'information enregistrée par le capteur $c_3$.*

Il existe d'autres configurations possibles :

- les trois capteurs enregistrent trois événements indépendants, $c_1$ et $c_2$ sont indépendants et indépendants conditionnellement à $c_3$ : $c_3$ n'apporte pas d'information rendant dépendants les deux autres capteurs.
- dans le cas de la figure 8, $c_2$ et $c_3$ sont dépendants et $c_2$ et $c_3$ sont dépendants conditionnellement à $c_1$ car $c_1$ ne contient pas d'information commune à $c_2$ et $c_3$.

Comme pour l'estimation de la structure de dépendance, l'estimation de la structure de dépendance conditionnelle peut être simplifiée à l'estimation de la structure d'indépendance conditionnelle. Nous sommes ainsi en présence d'un système binaire : soit les deux capteurs considérés sont indépendants conditionnellement au reste du réseau, soit ils ne le sont pas.

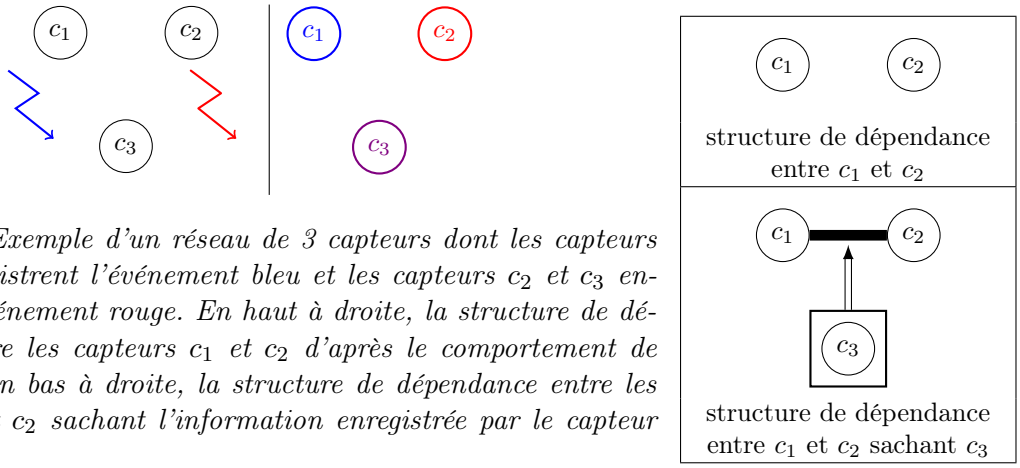



FIGURE *8 – Exemple d'un réseau de 3 capteurs dont les capteurs $c_1$ et $c_3$ enregistrent l'événement bleu et les capteurs $c_2$ et $c_3$ enregistrent l'événement rouge. En haut à droite, la structure de dépendance entre les capteurs $c_1$ et $c_2$ d'après le comportement de l'exemple et en bas à droite, la structure de dépendance entre les capteurs $c_1$ et $c_2$ sachant l'information enregistrée par le capteur $c_3$.*

Nous avons orienté nos travaux de recherche sur l'estimation de la structure d'indépendance conditionnelle d'un réseau de capteurs.

### Réseaux de capteurs = Processus multivariés gaussiens ?

Un processus multivarié est un processus regroupant plusieurs variables. Un réseau de capteurs peut être assimilé à un processus multivarié : chaque capteur correspond à une variable du processus. Nous transposons ainsi le problème d'estimation de la structure d'indépendance conditionnelle d'un réseau de capteurs à l'estimation de la structure d'indépendance conditionnelle d'un processus multivarié.

Nous choisissons de travailler avec des processus multivariés gaussiens. Ce choix est motivé par la simplicité de travailler avec de tels processus : pour ces processus, la notion de dépendance statistique se résume à la corrélation. Les seuls moments d'ordre deux sont alors nécessaires et suffisants pour notre étude. Dans le cas de signaux non gaussiens, les approches basées sur la notion de dépendance statistique requièrent l'utilisation de moments d'ordres supérieurs beaucoup plus difficile à estimer, surtout quand le nombre d'observations des processus est restreint. Concernant nos applications pratiques, nous étudions des processus réels qui subissent une transformée en ondelettes lors de l'étape de pré-traitement : la transformée en ondelettes est le produit scalaire entre une variable aléatoire et une fonction déterministe donc d'après le théorème central limite, les coefficients d'ondelettes à chaque échelle sont des variables aléatoires gaussiennes dans le cas asymptotique [Mou94]. Le fait que ces variables soient conjointement gaussiennes n'est cependant pas assuré. Dans l'ensemble de nos travaux, nous faisons néanmoins l'hypothèse que nos variables sont conjointement gaussiennes.

### Modèles graphiques gaussiens

En considérant que nous sommes en présence de processus multivariés gaussiens, notre problématique s'inscrit dans le cadre des modèles graphiques gaussiens. Un modèle graphique gaussien est un couple entre un processus $\mathbf{X}$ et un graphe $G$ : le graphe $G$ est la représentation de la structure d'indépendance conditionnelle du processus $\mathbf{X}$ (la figure 6 est un graphe). Le contexte des modèles graphiques gaussiens est utilisé aussi bien en génétique [TS14], en économie [GS13], en météorologie [ZFLH14] et dans de nombreux autres domaines.

## Objectifs

Dans le cadre de cette thèse, après avoir fait l'hypothèse d'être en présence de processus dont les variables sont conjointement gaussiennes, nous avons choisi de focaliser notre objectif sur l'estimation de modèles graphiques gaussiens associés à des processus donnés, pouvant être des réseaux de capteurs. La réalisation de cet objectif fut accompagné d'objectifs secondaires qui ont permis soit de comparer les performances de méthodes d'estimation de modèles graphiques gaussiens de façon rigoureuse, soit de donner un exemple d'utilisation d'une telle approche sur des données réelles.

## Organisation du manuscrit

Le présent manuscrit est décomposé en six chapitres. Ces chapitres ont été construits dans l'optique d'une lecture linéaire du manuscrit.

Le chapitre 1, *Modèles Graphiques Gaussiens*, nous permet de présenter le concept de modèles graphiques gaussiens notamment en explicitant la notion d'indépendance conditionnelle et en introduisant les définitions de la théorie des graphes nécessaires à la bonne compréhension de la notion de modèles graphiques gaussiens ainsi que des méthodes pour les estimer. Après avoir mis en lumière ce concept, nous dressons un portrait de différentes méthodes utilisées pour estimer le modèle graphique gaussien associé à un processus, en mettant en avant leurs atouts et leurs inconvénients. A l'issue de ce chapitre, nous identifions les besoins auxquels les méthodes actuelles ne répondent pas afin de développer une méthode répondant à ces besoins.

Le chapitre 2, *Procédure d'évaluation de méthodes d'estimation de modèles graphiques gaussiens*, a pour objectif de mettre en place des outils permettant de valider les performances de notre méthode mais également des méthodes existantes. Ces outils permettent aussi d'évaluer si notre méthode répond aux besoins identifiés. Les outils proposés sont : une méthode de simulation de processus multivariés gaussiens dont la structure d'indépendance conditionnelle est connue et des métriques pour évaluer les performances de la méthode sachant la structure d'indépendance conditionnelle attendue. Ces outils sont proposés après avoir fait un bilan des outils existants.

Le chapitre 3, *Méthode ABiGlasso*, présente la méthode que nous avons développée. Cette méthode a pour objectif de comparer des graphes candidats à représenter la structure d'indépendance conditionnelle d'un processus en un temps acceptable. Cette méthode se décline en plusieurs variantes. Nous testons les performances en terme de temps de calcul et d'estimation de la structure d'indépendance conditionnelle de ces variantes afin d'offrir la combinaison entre variante et paramètres d'entrée qui donne le meilleur compromis entre les performances d'estimation et le temps de calcul.

Le chapitre 4, *Études comparatives de méthodes d'estimation de modèles graphiques gaussiens*, situe notre méthode parmi les méthodes existantes suivant différents critères comme le temps de calcul, les performances d'estimation mais également la capacité à fournir un score sur les graphes étudiés ou sur les arêtes.

Le chapitre 5, *Étude conjointe de processus à l'aide de leurs structures d'indépendance conditionnelle*, présente une nouvelle approche permettant de comparer des processus en utilisant leurs structures d'indépendance conditionnelle. Elle permet plus exactement de classifier des groupes de processus ayant des structures d'indépendance conditionnelle différentes dans l'objectif d'utiliser cette approche pour la classification de données réelles.

Le chapitre 6, *Application dans le contexte de la connectivité fonctionnelle cérébrale*, présente dans un premier temps le principe de l'IRM (imagerie par résonance magnétique) fonctionnelle et comment est étudiée la connectivité fonctionnelle cérébrale à partir de telles données. Dans un deuxième temps, nous appliquons l'approche du chapitre 5 à l'identification d'ensembles de régions cérébrales pertinents pour mener des études sur des patients dans le coma.

Nous terminons le manuscrit par un tour d'horizon des perspectives ouvertes ou mises en avant par les travaux menés dans le cadre de cette thèse.

## Matériel complémentaire

Une Toolbox Matlab a été développée à partir des travaux de cette thèse. Elle regroupe différentes méthodes d'estimation de la structure d'indépendance conditionnelle d'un processus gaussien multivarié ainsi que d'autres éléments décrits dans ce manuscrit. Chaque fois qu'un élément décrit dans le manuscrit est présent dans la Toolbox, nous le signalerons. Cette Toolbox est pour le moment disponible à l'adresse suivante : http://www.gipsa-lab.fr/~aude.costard/toolbox.html

# Chapitre 1

# Modèles Graphiques Gaussiens

**Sommaire**



Nous travaillons sur des réseaux de capteurs et nous souhaitons extraire leur structure d'indépendance conditionnelle. La structure d'indépendance conditionnelle d'un réseau de capteurs permet de savoir quels capteurs sont indépendants sachant l'information portée par les autres capteurs du réseau. Nous nous intéressons uniquement aux relations instantanées, c'est-à-dire que nous ne prenons en compte que l'information portée simultanément par les différents capteurs. Ce choix repose sur le fait que nous considérons des réseaux de capteurs pour lesquels le temps d'intégration, c'est-à-dire le temps entre deux échantillons, est grand devant le temps de propagation des phénomènes physiques enregistrés. Nous assimilons nos réseaux de capteurs à des processus multivariés : chaque variable du processus correspond à un capteur du réseau. De

plus, nous considérons que ces processus sont multivariés gaussiens. Nous supposons également que nos processus sont ergodiques. Les modèles graphiques gaussiens associent un graphe et un processus multivarié gaussien, le graphe étant la représentation de la structure d'indépendance conditionnelle du processus. Pour estimer la structure d'indépendance conditionnelle d'un réseau de capteurs, nous cherchons donc à estimer le modèle graphique gaussien associé au processus multivarié gaussien qu'est notre réseau. Il existe diverses méthodes pour estimer des modèles graphiques gaussiens, chacune ayant ses propres avantages et inconvénients.

Dans ce chapitre, nous présentons les modèles graphiques gaussiens et les méthodes qui existent pour les estimer. Dans une première partie, nous introduisons l'indépendance conditionnelle, la loi de Wishart et quelques éléments de la théorie des graphes nous permettant de définir un modèle graphique gaussien. Ces notions servent également à mieux comprendre les méthodes d'estimation des modèles graphiques gaussiens qui sont présentées dans la deuxième partie du chapitre, avec leurs avantages et leurs inconvénients.

## 1.1 Notions de base

Dans cette partie nous introduisons différentes notions et définitions nécessaires à la bonne compréhension des modèles graphiques gaussiens ainsi qu'aux méthodes servant à estimer ces modèles. Cette section est basée sur les livres de Whittaker [Whi90], de Lauritzen [Lau96] et Anderson [And03] où le lecteur pourra trouver une présentation exhaustive des modèles graphiques gaussiens ou des lois multivariées. Nous présentons ainsi l'indépendance conditionnelle, des définitions de la théorie des graphes, les lois de Wishart et la notion de parcimonie.

### 1.1.1 Indépendance conditionnelle

Pour pouvoir étudier la structure d'indépendance conditionnelle d'un processus, il faut comprendre ce que représente l'indépendance conditionnelle entre deux variables. Dans cette section, ainsi que dans l'ensemble du manuscrit, nous travaillons avec des processus multivariés $\mathbf{X}$ qui sont assimilables à des matrices $n \times p$, $n$ étant le nombre d'observations du processus (ce qui équivaut au nombre d'échantillons de la série temporelle) et $p$ son nombre de variables (équivalent aux nombres de capteurs dans le cas de réseaux de capteurs). $X_i$ désigne la $i^e$ colonne de $\mathbf{X}$, c'est-à-dire la $i^e$ variable du processus et $X_i$ est de longueur $n$.

#### 1.1.1.1 Définition

**Définition 1.1.1.** ***Indépendance statistique***

*Soient $X$ et $Y$ deux variables aléatoires. $X$ et $Y$ sont indépendantes si et seulement si $\mathbb{P}_{XY}(x, y) = \mathbb{P}_X(x)\mathbb{P}_Y(y)$ où $\mathbb{P}_X(x)$ est la probabilité que la variable $X$ prenne la valeur $x$ et $\mathbb{P}_{XY}(x, y)$ est la probabilité conjointe de $X$ et de $Y$, c'est-à-dire la probabilité que $X = x$ et $Y = y$.*

L'indépendance entre $X$ et $Y$ est notée $X \perp\!\!\!\perp Y$.

Dans le cas gaussien, deux variables sont indépendantes si leur corrélation est nulle.

Cette forme d'indépendance ne prend pas en compte l'influence des autres variables du processus auquel appartiennent les deux variables étudiées. Pour prendre en compte cette influence nous avons recours à l'indépendance conditionnelle

**Définition 1.1.2.** ***Indépendance conditionnelle***

*Soient $X$, $Y$ et $Z$ trois variables aléatoires, $X$ et $Y$ sont indépendantes conditionnellement à $Z$ si et seulement si $\mathbb{P}_{XY|Z}(x, y|z) = \mathbb{P}_{X|Z}(x|z)\mathbb{P}_{Y|Z}(y|z)$ où $\mathbb{P}_{X|Z}(x|z)$ est la probabilité que $X = x$ sachant que $Z = z$.*

L'indépendance entre $X$ et $Y$ conditionnellement à $Z$ est notée $X \perp\!\!\!\perp Y|Z$.

Dans le cas d'un processus multivarié, $Z$ n'est pas une unique variable mais l'ensemble des variables du processus à l'exception des variables $X$ et $Y$.

Se pose maintenant la question de savoir comment mesurer l'indépendance conditionnelle.

#### 1.1.1.2 Corrélation partielle

Nous sommes en présence de processus multivariés supposés gaussiens et ergodiques. L'indépendance instantanée entre deux variables d'un processus correspond à une corrélation nulle entre ces deux variables. Or, nous nous intéressons à l'indépendance conditionnelle instantanée entre ces deux variables.

Soient deux variables $X$ et $Y$, soit $\widehat{X}(Y)$ l'estimation linéaire de $X$ sachant $Y$, $\widehat{X}(Y) = \beta_{XY}Y$. Le résidu de $X$ sachant $Y$ vaut $\epsilon_X = X - \widehat{X}(Y)$ et $\beta_{XY}$ correspond au coefficient de régression de $X$ sachant $Y$.

**Définition 1.1.3.** ***Corrélation partielle***

*La corrélation partielle de deux variables $X$ et $Y$ sachant une troisième variable $Z$ est la corrélation des résidus de $X$ et $Y$ sachant $Z$ : $cor(X, Y|Z) = cor(X - \widehat{X}(Z), Y - \widehat{Y}(Z))$*

Soit $\mathbf{X}$ un processus gaussien multivarié dont les composantes sont indexées par l'ensemble $V$. Nous notons $X_{V\setminus\{i,j\}}$ l'ensemble des variables de $\mathbf{X}$ privé des variables $X_i$ et $X_j$. Soit $\widehat{X}_i(X_{V\setminus\{i,j\}})$ l'estimation linéaire de $X_i$ sachant $X_{V\setminus\{i,j\}}$ :

$$\widehat{X}_i(X_{V\setminus\{i,j\}}) = \sum_{k\in V\setminus\{i,j\}} \beta_{ik}X_k.$$

$\epsilon_{ij} = X_i - \widehat{X}_i(X_{V\setminus\{i,j\}})$ est le résidu de $X_i$ sachant $X_{V\setminus\{i,j\}}$.

Pour un processus gaussien multivarié, l'indépendance conditionnelle entre deux composantes conditionnellement aux autres composantes du processus est équivalente à une corrélation nulle entre les résidus $\epsilon_{ij}$ et $\epsilon_{ji}$, c'est-à-dire une corrélation partielle nulle entre $X_i$ et $X_j$ sachant $X_{V\setminus\{i,j\}}$.

**Calcul de la corrélation partielle**

Il est difficile de calculer les résidus d'estimation d'une variable en fonction d'un groupe de variables, principalement quand le nombre de variables à considérer est important. Whittaker [Whi90] a montré que calculer la matrice des corrélations partielles pouvait se faire en utilisant l'inverse de la covariance.

Soient $\Sigma$ la matrice de covariance de $\mathbf{X}$ et $K = \Sigma^{-1}$ sa matrice de précision, la matrice des corrélations partielles $\Pi$ se calcule de la façon suivante :

$$\pi_{ij} = \begin{cases} -\frac{k_{ij}}{\sqrt{k_{ii}k_{jj}}} & \text{si } i \neq j \\ 1 & \text{sinon} \end{cases} \tag{1.1}$$

Notons que $\Sigma$ étant la matrice de covariance de $\mathbf{X}$ elle est par définition définie positive et donc toujours inversible.

Par définition $\pi_{ij} = 0 \Leftrightarrow k_{ij} = 0$. Donc pour un processus gaussien multivarié $\mathbf{X}$ :

$$X_i \perp\!\!\!\perp X_j|X_{V\setminus\{i,j\}} \Leftrightarrow k_{ij} = 0. \tag{1.2}$$

**Illustration de la différence entre corrélation et corrélation partielle**

La corrélation et la corrélation partielle rendent compte de deux phénomènes différents. Les deux exemples ci-dessous illustrent ces différences de comportement.

*Exemple 1 :* Soit un processus multivarié gaussien de composantes $X$, $Y$ et $Z$ définies ainsi :

$$X \sim \mathcal{N}(0,1), \quad Y = \frac{5X + \eta_1}{6}, \eta_1 \sim \mathcal{N}(0,1), \quad Z = \frac{5Y + \eta_2}{6}, \eta_2 \sim \mathcal{N}(0,1)$$

où $\mathcal{N}(\mu, \sigma)$ est la loi gaussienne de moyenne $\mu$ et de variance $\sigma$. Dans cet exemple simple, le calcul exact est trivial. Dans la suite nous présentons des résultats de simulations sur 100 tirages aléatoires des lois de $X$, $Y$ et $Z$. Le calcul de la matrice de corrélation empirique et de la matrice des corrélations partielles empirique de ce processus donne le résultat suivant :

*Corrélation*

| | $X$ | $Y$ | $Z$ |
|---|---|---|---|
| $X$ | 1 | 0.98 | 0.95 |
| $Y$ | 0.98 | 1 | 0.97 |
| $Z$ | 0.95 | 0.97 | 1 |

*Corrélation partielle*

| | $X$ | $Y$ | $Z$ |
|---|---|---|---|
| $X$ | 1 | 0.80 | 0.02 |
| $Y$ | 0.80 | 1 | 0.57 |
| $Z$ | 0.02 | 0.57 | 1 |

Cet exemple montre que les composantes $X$ et $Z$ sont très dépendantes car leur corrélation vaut 0.95 mais qu'elles sont indépendantes conditionnellement à $Y$ car leur corrélation partielle vaut 0.02 (bien qu'elle ne soit pas égale à 0, elle est statistiquement significativement nulle car nous avons un nombre fini d'observations).

△

*Exemple 2 :* Soit un processus multivarié gaussien de composantes $X$, $Y$ et $Z$ définies ainsi :

$$X \sim \mathcal{N}(0,1), \quad Y \sim \mathcal{N}(0,1), \quad Z = \frac{5X + 5Y + \eta}{11}, \eta \sim \mathcal{N}(0,1)$$

où $\mathcal{N}(\mu, \sigma)$ est la loi gaussienne de moyenne $\mu$ et de variance $\sigma$. Le calcul de la matrice de corrélation empirique et de la matrice des corrélations partielles empirique de ce processus donne le résultat suivant :

*Corrélation*

| | $X$ | $Y$ | $Z$ |
|---|---|---|---|
| $X$ | 1 | 0.04 | 0.75 |
| $Y$ | 0.04 | 1 | 0.68 |
| $Z$ | 0.75 | 0.68 | 1 |

*Corrélation partielle*

| | $X$ | $Y$ | $Z$ |
|---|---|---|---|
| $X$ | 1 | -0.97 | 0.99 |
| $Y$ | -0.97 | 1 | 0.98 |
| $Z$ | 0.99 | 0.98 | 1 |

Cet exemple montre que les composantes $X$ et $Y$ sont indépendantes car leur corrélation vaut 0.04 mais qu'elles sont très dépendantes conditionnellement à $Z$ car leur corrélation partielle vaut -0.97.

△

#### 1.1.1.3 Indépendance conditionnelle et régression linéaire

Pour savoir si deux variables d'un processus multivarié gaussien sont indépendantes conditionnellement au reste du processus, il est possible d'utiliser la régression linéaire.

**Théorème 1.1.4.** *Soit un processus multivarié gaussien* $\mathbf{X}$*, l'estimateur linéaire de* $X_i$ *sachant les autres variables du processus est* $\widehat{X}_i(X_{V\setminus\{i\}}) = \beta_{ij}X_j + \sum_{k\in V\setminus\{i,j\}} \beta_{ik}X_k$ *où les* $\beta_{ik}$ *sont les coefficients de régression. Dire que* $X_i$ *et* $X_j$ *sont indépendantes conditionnellement à* $X_{V\setminus\{i,j\}}$ *équivaut à avoir* $\beta_{ij} = 0$*.*

La démonstration de ce théorème est donné dans l'annexe A.

Nous avons vu que l'indépendance conditionnelle entre deux composantes, dans le cas de processus gaussiens multivariés, peut être obtenue en calculant la corrélation partielle de ces deux composantes. Elle peut aussi être obtenue en estimant les coefficients de régressions linéaires. Ces deux propriétés sont utilisées dans les méthodes d'estimation de modèles graphiques gaussiens que nous verrons dans la deuxième partie de ce chapitre (1.2).

### 1.1.2 Matrice de covariance empirique et distribution de Wishart

Dans cette section, nous présentons la distribution de Wishart qui est la distribution que suit une matrice de covariance empirique [And03]. Cette distribution a été créée à partir de travaux de Wishart [Wis28]. Nous présentons également la distribution de Wishart inverse qui est la distribution suivie par une matrice de précision empirique. Ces lois sont utilisées dans certaines méthodes d'estimation de modèles graphiques gaussiens que nous verrons dans la deuxième partie de ce chapitre comme par exemple la méthode du Bayesian Adaptive Glasso (section 1.2.2.2) ou les méthodes bayésiennes (section 1.2.4).

#### 1.1.2.1 Distribution de Wishart

**Définition 1.1.5.** ***Distribution de Wishart***

*Soit* $\mathbf{X}$ *un processus multivarié gaussien de* $p$ *variables et* $n$ *observations, de moyenne nulle et de covariance* $\Sigma$ *(*$\mathbf{X} \sim \mathcal{N}_p(0, \Sigma)$*) et* $M = \mathbf{X}^T\mathbf{X}$*. La distribution de Wishart est la distribution de probabilité de la matrice* $M$ *de taille* $p \times p$ *:*

$$M \sim \mathcal{W}_p(n, \Sigma)$$

*où* $n$ *est le nombre de degrés de liberté de la distribution et* $\Sigma$ *sa matrice d'échelle.*

La densité de probabilité d'une distribution de Wishart de $n$ degrés de liberté et de matrice d'échelle $\Sigma$, pour la matrice $M$, vaut :

$$\mathbb{P}(M) = \frac{|M|^{\frac{n-p-1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\Sigma^{-1}M)\right)}{2^{\frac{np}{2}}|\Sigma|^{\frac{n}{2}}\Gamma_p\left(\frac{n}{2}\right)}$$

où $|M|$ désigne le déterminant de $M$, tr est la fonction trace et $\Gamma_p$ est la fonction gamma multivariée :

$$\Gamma_p(a) = \pi^{p(p-1)/4} \sum_{j=1}^{p} \Gamma\left[a + \frac{1-j}{2}\right]..$$

On définit la constante de Wishart : $\omega(p, n) = 2^{\frac{np}{2}}\Gamma_p\left(\frac{n}{2}\right)$.
La matrice de covariance empirique $\Sigma^{emp}$ vaut $\frac{M}{n}$ et suit la distribution $\mathcal{W}_p(n, \frac{\Sigma}{n})$.

#### 1.1.2.2 Distribution de Wishart inverse

**Définition 1.1.6.** ***Distribution de Wishart inverse***
*Soit $M \sim \mathcal{W}_p(n, \Sigma)$, la distribution de Wishart inverse est la distribution de probabilité de la matrice $M^{-1}$ de taille $p \times p$ :*

$$M^{-1} \sim \mathcal{IW}_p(n, \Psi)$$

où $\Psi = \Sigma^{-1}$.

La densité de probabilité d'une distribution de Wishart inverse de $n$ degrés de liberté et de la matrice d'échelle $\Psi$, pour la matrice $M^{-1}$ vaut :

$$\mathbb{P}(M^{-1}) = \omega(p,n)^{-1}|\Psi|^{\frac{n}{2}}|M^{-1}|^{-\frac{n+p+1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\Psi M)\right).$$

La matrice de précision empirique $K^{emp}$ est l'inverse de la matrice de covariance empirique $\Sigma^{emp}$ et donc suit la distribution $\mathcal{IW}_p(n, \frac{\Sigma^{-1}}{n})$.

### 1.1.3 Éléments de théorie des graphes

Dans cette section nous introduisons les notations et les notions utilisées dans la théorie des graphes nécessaires à une meilleure compréhension des modèles graphiques gaussiens et de leurs méthodes d'estimation.

#### 1.1.3.1 Graphes

**Définition 1.1.7.** ***Graphe***
*Un graphe $G$ est un ensemble fini de nœuds $V$ reliés par des arêtes. L'ensemble des arêtes est noté $E$ et $E \subset V \times V$. On note $G = (V, E)$.*

Quand il est évident que les ensembles $E$ et $V$ sont associés au graphe $G$, la notation $E$ et $V$ est utilisée telle quelle. Si on parle de plusieurs graphes, $G_1$ et $G_2$ par exemple, nous utilisons les notations $E(G_1)$, $E(G_2)$, $V(G_1)$ et $V(G_2)$.

Comme discuté dans l'introduction, nous étudions des interactions non-dirigées, nous travaillons donc uniquement avec des graphes non-dirigés.

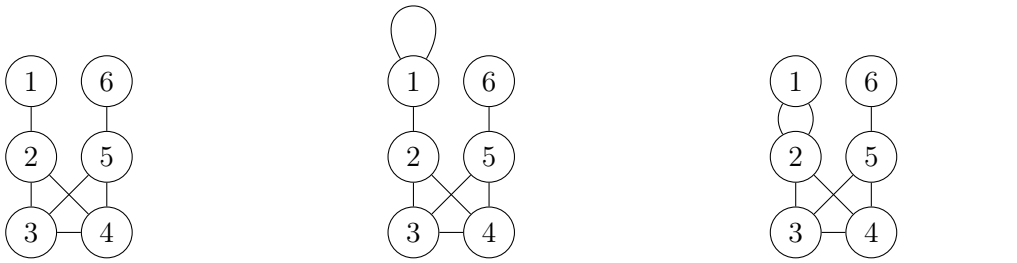
**Définition 1.1.8.** ***Graphe non-dirigé***
*Un graphe non dirigé est un graphe pour lequel les arêtes ne sont pas dirigées. Cela signifie que l'arête $(i, j)$ est équivalente à l'arête $(j, i)$.*

Comme nos graphes ne sont pas dirigés, un nœud ne peut pas être couplé avec lui-même et il existe une unique façon de coupler deux nœuds. Ce sont des graphes simples.

**Définition 1.1.9.** ***Graphe simple***
*Un graphe simple est un graphe sans boucle, c'est-à-dire sans arêtes reliant un nœud à lui-même, ni arêtes multiples.*

La figure 1.1 donne des exemples de graphes non-dirigés, le premier étant simple, les autres ne l'étant pas.

Pour le graphe (a) de la figure 1.1 :
- $V = \{1, 2, 3, 4, 5, 6\}$
- $E = \{(1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (4, 5), (5, 6)\}$.




(a) ✓ (b) ✗ : boucle sur 1 (c) ✗ : arête multiple entre 1 et 2

FIGURE *1.1 – Exemples de graphes non-dirigés : le premier est un graphe non-dirigé simple, les deux autres ne respectent pas la définition donnée précédemment. Le graphe (b) a une boucle sur le nœud 1 et le graphe (c) a une double arête entre les nœuds 1 et 2.*

Les graphes non-dirigés peuvent être représentés sous la forme de matrices binaires, ce sont les matrices d'adjacences.

**Définition 1.1.10.** ***Matrice d'adjacence***

*La matrice d'adjacence $A$ d'un graphe $G$ est construite de la façon suivante :*

$$\begin{cases} a_{ij} = 1 & si\ (i,j) \in E \\ a_{ij} = 0 & sinon \end{cases}$$

Nous choisissons de définir que $a_{ii} = 0$ comme le graphe est simple. Le graphe (a) de la figure 1.1 peut être représenté ainsi :

$$A = \begin{array}{c} \begin{array}{cccccc} {\scriptstyle 1} & {\scriptstyle 2} & {\scriptstyle 3} & {\scriptstyle 4} & {\scriptstyle 5} & {\scriptstyle 6} \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{array}{l} {\scriptstyle 1} \\ {\scriptstyle 2} \\ {\scriptstyle 3} \\ {\scriptstyle 4} \\ {\scriptstyle 5} \\ {\scriptstyle 6} \end{array}$$

Pour un graphe non-dirigé, $A$ est symétrique : puisque $(i,j) = (j,i)$, $a_{ij} = a_{ji}$.

Dans le cas des graphes non-dirigés, il est également possible de représenter un graphe par un vecteur composé de 0 pour les arêtes non-présentes et de 1 pour les arêtes présentes. Afin que ce vecteur soit reproductible, nous ordonnons les arêtes de la façon suivante :

- le couple désignant une arête $(i,j)$ est tel que $i \leq j$
- si $i_a < i_b$ et $j_a < j_b$ alors les arêtes sont ordonnées ainsi : $(i_a, j_a)$, $(i_a, j_b)$, $(i_b, j_a)$, $(i_b, j_b)$.

**Définition 1.1.11.** ***Support de graphe :*** *vecteur représentant un graphe non-dirigé, après ordonnancement des arêtes. Une arête présente est représentée par un 1 et une arête non-présente par un 0. Notons que comme nous considérons uniquement des graphes simples, les arêtes de type boucle correspondant à la diagonale de la matrice d'adjacence sont exclues du support de graphe.*

Le support de graphe correspond à la concaténation des lignes de la partie triangulaire supérieure de la matrice d'adjacence. Pour le graphe (a) de la figure 1.1, le support de graphe est : [100001100110101].

### 1.1.3.2 Décomposition de graphes

Il est possible de décomposer un graphe ayant un grand nombre de nœuds en plusieurs graphes de tailles inférieures. Cette section introduit la notion de décomposition de graphes et la définition d'un graphe décomposable dont les propriétés que nous verrons par la suite sont utilisées par certaines méthodes d'estimation des modèles graphiques gaussiens. Pour comprendre cette notion de décomposabilité d'un graphe, nous avons besoin de quelques définitions.

**Définition 1.1.12.** ***Nœuds adjacents***
*Deux nœuds d'un même graphe sont dits adjacents si ils sont reliés par une arête.*

L'ensemble des nœuds adjacents à un nœud $a$ est noté $\text{adj}(a)$. Par exemple, pour le graphe (a) de la figure 1.1, $\text{adj}(2) = \{1, 3, 4\}$.

**Définition 1.1.13.** ***Chemin***
*Un chemin de longueur $k$ entre les nœuds $a$ et $b \in V$ est une séquence $a = a_0, a_1, \ldots, a_k = b$ de $k+1$ nœuds distincts tels que $(a_i, a_{i+1}) \in E \; \forall i = 0, \ldots, k-1$.*

La figure 1.2 illustre la possible non unicité d'un chemin reliant les mêmes nœuds et ayant une longueur donnée. Elle donne deux exemples de chemins de longueur 4 reliant les nœuds 1 et 6 du graphe (a) de la figure 1.1.


(a) chemin 1 (b) chemin 2

FIGURE *1.2 – Exemples de chemins de longueur 4 entre les nœuds 1 et 6 du graphe (a) de la figure 1.1.*

**Définition 1.1.14.** ***Cycle***
*Un cycle est un chemin dont le nœud de départ est le même que le nœud d'arrivée.*

**Définition 1.1.15.** ***Séparateur***
*Soient $A$, $B$ et $S$ trois sous-ensembles de $V$, $S$ est un séparateur de $A$ et $B$ dans $G$ si et seulement si tout chemin reliant un nœud de $A$ à un nœud de $B$ passe par un nœud de $S$. Cette relation est notée $A \perp\!\!\!\perp_G B | S$.*

Pour le graphe (a) de la figure 1.1, $\{1, 2\} \perp\!\!\!\perp_G \{6\} | \{3, 4\}$ (cf figure 1.3).

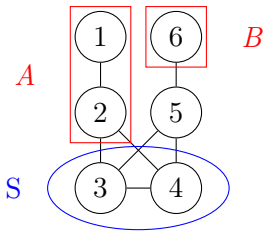



FIGURE *1.3 – Exemple de séparateur pour le graphe (a) de la figure 1.1 :* $\{1, 2\} \perp\!\!\!\perp_G \{6\} | \{3, 4\}$

**Définition 1.1.16.** ***Sous-graphe***
*Le sous-graphe de $G$ induit par $A$ est le graphe $G_A = (A, E_A)$ où $E_A = E \cap (A \times A)$.*

**Définition 1.1.17.** ***Graphe complet***
*Un graphe complet est un graphe dont tous les nœuds sont reliés deux à deux.*

Un graphe complet avec $p$ nœuds a $\frac{p(p-1)}{2}$ arêtes.

**Définition 1.1.18.** ***Clique***
*Soit $G_A$ un sous-graphe complet de $G$, si pour tout nœud $i \in V \setminus A$, le sous-graphe $G_{A\cup\{i\}}$ de $G$ n'est pas complet, alors $G_A$ est une* ***clique*** *de $G$.*

L'ensemble des cliques d'un graphe est noté $\mathcal{C}$ et cet ensemble est unique.
Par exemple $(2, 3, 4)$ est une clique du graphe (a) de la figure 1.1 car $(2, 3), (2, 4)$ et $(3, 4)$ appartiennent à $E$ et si le nœud 1, 5 ou 6 est ajouté, le graphe obtenu n'est pas complet ($(1, 4) \notin E$, $(2, 5) \notin E$ et $(6, 3) \notin E$).

Soient $A$ et $B$ deux ensembles, la notation $A \setminus B$ désigne l'ensemble $A$ privé de $A \cap B$.

**Définition 1.1.19.** ***Décomposition d'un graphe***
*La paire $(A, B)$ de sous-ensembles de $V$ est une décomposition de $G$ si et seulement si $A \cap B$ est un séparateur de $A \setminus B$ et $B \setminus A$ et est complet.*

Seules les décompositions propres sont considérées dans la suite, c'est-à-dire les décompositions pour lesquelles $A \setminus B \neq \emptyset$ et $B \setminus A \neq \emptyset$.

**Définition 1.1.20.** ***Composante primaire***
*Une composante primaire est un sous-graphe n'admettant pas de décomposition.*

Un graphe peut être décomposé en composantes primaires et cette décomposition est unique.

**Définition 1.1.21.** ***Graphe décomposable***
*Un graphe décomposable est un graphe dont les composantes primaires sont toutes complètes, c'est-à-dire que sa décomposition est composée uniquement de cliques.*

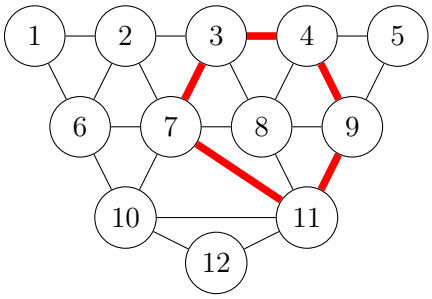
Pour un graphe $G$ décomposable, $\mathcal{S}$ correspond à l'ensemble des séparateurs séparant les $k$ cliques de $G$. $\mathcal{S}$ est de taille $k - 1$.

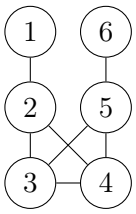
La figure 1.4 représente deux graphes qui diffèrent d'une seule arête. Cependant l'un est décomposable, l'autre est non-décomposable.
Un graphe décomposable est aussi appelé graphe cordal.

**Définition 1.1.22.** ***Corde***
*Une corde est une arête reliant deux nœuds non-successifs d'un cycle.*

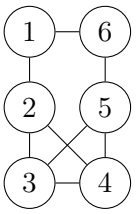
Un graphe décomposable est aussi appelé graphe cordal car un graphe est décomposable si et seulement si il n'a pas de cycles de longueur supérieure ou égale à 4 qui n'aient pas de corde.

*Exemple :* (d'après le cours de Roverato, école d'été GMINeuro, Grenoble 2013)
Ce graphe est-il décomposable ?






(a) graphe décomposable




(b) graphe non-décomposable

FIGURE *1.4 – Exemple d'un graphe décomposable et d'un graphe non décomposable qui diffèrent d'une arête :* $(1,6)$*. La décomposition en composantes primaires du premier graphe est* $(1,2),(2,3,4),(3,4,5)$ *et* $(5,6)$ *qui sont toutes des sous-graphes complets. La décomposition en composantes primaires du second graphe est* $(2,3,4),(3,4,5)$ *et* $(1,2,3,5,6)$*, la dernière composante n'est pas un sous-graphe complet (par exemple* $(1,5) \notin E$*).*

Non car le cycle en rouge est de longueur 5 et n'a pas de corde : aucun des nœuds non-consécutifs du cycle ne sont reliés entre eux.

△

Il est possible de rendre un graphe décomposable en le triangularisant.

**Définition 1.1.23.** ***Triangulariser un graphe***

*Triangulariser un graphe revient à ajouter des arêtes afin de supprimer les cycles de longueur supérieure ou égale à 4 ayant des nœuds non consécutifs sans corde.*

Le graphe rendu décomposable est appelé graphe triangulé.

**Intérêt des graphes décomposables**

Les propriétés de Markov sont vérifiées sur les graphes décomposables : soit $S$ un séparateur de $A$ et $B$ deux sous-ensembles de $V$ alors $X_A \perp\!\!\!\perp X_B | X_S$. De plus comme nous travaillons avec des processus gaussiens multivariés, la fonction de densité de $\mathbf{X}$ est strictement positive. Les propriétés de Markov et la densité strictement positive font que la distribution de $\mathbf{X}$ se factorise.

Dans le cas des graphes décomposables, la distribution de $\mathbf{X}$ se factorise sous la forme suivante :

$$\mathbb{P}(\mathbf{X}) = \frac{\prod_{i=1}^{|\mathcal{C}|} \mathbb{P}_{C_i}(X_{C_i})}{\prod_{j=1}^{|\mathcal{S}|} \mathbb{P}_{S_j}(X_{S_j})}$$

Cette propriété de factorisation permet de simplifier les calculs dans le cas de graphes décomposables, c'est pour cela que certaines méthodes d'estimation que nous verrons dans la suite se limitent aux graphes décomposables.

### 1.1.4 Modèles graphiques gaussiens

Soient $G = (V, E)$ un graphe à $p$ nœuds et $\mathbf{X}$ un processus gaussien multivarié tel que $\mathbf{X} \sim \mathcal{N}_p(\mathbf{0}, \Sigma)$.

**Définition 1.1.24.** ***Modèle Graphique Gaussien :*** $(G, \mathbf{X})$ *est un modèle graphique gaussien signifie :*

$$\begin{gathered} i \in V : \textit{variable } X_i \\ (i,j) \notin E \Leftrightarrow X_i \perp\!\!\!\perp X_j | X_{V \setminus \{i,j\}} \Leftrightarrow k_{ij} = 0 \end{gathered} \tag{1.3}$$

Cela revient à dire que chaque variable de $\mathbf{X}$ est associée à un nœud de $G$ et que deux nœuds ne sont pas reliés par une arête si et seulement si les deux variables associées aux nœuds sont indépendantes conditionnellement aux autres variables, c'est-à-dire si leur corrélation partielle (précision) est nulle.

### 1.1.5 Parcimonie

Les données ayant un grand nombre de valeurs nulles sont souvent qualifiées de données parcimonieuses. Cela s'applique aussi bien aux matrices ayant un grand nombre de valeurs nulles qu'aux graphes ayant peu d'arêtes : on parle alors de matrices parcimonieuses et de graphes parcimonieux.

La définition des modèles graphiques gaussiens (définition 1.1.24) implique que lorsque deux variables sont indépendantes conditionnellement aux autres, elles ne sont pas reliées par une arête dans $G$ et leur coefficient dans la matrice de précision est nul. Nous sommes en présence de matrices avec des valeurs nulles et de graphes ayant certaines arêtes absentes. La question qui se pose est de savoir si on peut qualifier ces matrices et graphes de parcimonieux.

Il n'existe pas de définition univoque de la parcimonie.

Pour la parcimonie dans les graphes, Meinshausen et Bühlmann [MB06b] proposent différentes définitions :

- le degré $d$ d'un nœud, c'est-à-dire le nombre de nœuds auxquels il est relié, est limité : $d << p - 1$.
- en utilisant l'approche équivalente par régression linéaire, une différente alternative est de limiter le nombre de coefficients non nuls associés à deux nœuds reliés par une arête.

Ces définitions de la parcimonie imposent une certaine organisation au graphe considéré. Nous sommes dans un contexte où nous cherchons à estimer la structure d'indépendance conditionnelle d'un réseau de capteurs, sans rien imposer sur cette structure. Nous ne pouvons donc pas suivre les définitions de la parcimonie données. Dans le cadre de nos travaux nous utilisons la définition suivante :

**Définition 1.1.25.** ***Matrice parcimonieuse*** *Matrice ayant des valeurs nulles hors diagonale.*

Un graphe parcimonieux est un graphe dont la matrice d'adjacence est parcimonieuse. Nous introduisons également la notion de degré de parcimonie.

**Définition 1.1.26.** ***Degré de parcimonie*** *Indice permettant de quantifier la quantité de valeurs nulles hors diagonale d'une matrice parcimonieuse de taille $p \times p$.*

$$d_0 = \frac{\textit{nombre de valeurs nulles}}{p(p-1)}.$$

Dans la suite de ce manuscrit, nous désignons par matrices avec un fort degré de parcimonie les matrices ayant un grand nombre de valeurs nulles par rapport au nombre total de valeurs dans la matrice. Cela signifie que le degré de parcimonie est proche de 1, sa valeur maximale.

## 1.2 Estimation

Il existe de nombreuses méthodes pour estimer la structure d'indépendance conditionnelle d'un processus multivarié gaussien. Elles reposent toutes sur la définition des modèles graphiques gaussiens (définition 1.1.24) selon laquelle : $(i,j) \notin E \Leftrightarrow X_i \perp\!\!\!\perp X_j | X_{V\setminus\{i,j\}} \Leftrightarrow k_{ij} = 0$. Dempster [Dem72] fut le premier à vouloir estimer la matrice de covariance en imposant que certaines valeurs de son inverse soient nulles, introduisant le problème de *sélection de covariance*. Le

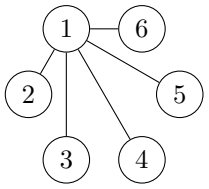
problème de *sélection de covariance* consiste à chercher la matrice de covariance qui soit la plus proche possible de la matrice de covariance empirique tout en ayant un maximum de valeurs nulles dans son inverse. Nous rappelons que l'inverse de la matrice de covariance est appelée matrice de précision. Le problème de *sélection de covariance* peut être aussi défini comme la recherche d'une matrice de précision avec un fort degré de parcimonie, c'est-à-dire avec un nombre important de valeurs nulles (cf définition 1.1.25), dont l'inverse est proche de la matrice de covariance empirique. Wermuth [Wer76] a montré par la suite que les valeurs non nulles de la matrice de précision correspondent aux arêtes d'un graphe représentatif des données étudiées dans le cadre de processus gaussiens multivariés. Ce sont ces travaux qui ont donné naissance à la définition des modèles graphiques gaussiens (définition 1.1.24).

Il existe différentes façons d'aborder l'estimation de la structure d'indépendance conditionnelle dans le cadre des modèles graphiques gaussiens mais toutes se basent sur la relation (1.3). La diversité des approches s'explique par le fait que chaque méthode a ses propres avantages et surtout ses propres inconvénients. Nous explicitons ces différents aspects dans les paragraphes suivants en ordonnant les méthodes de la façon suivante :

1. les méthodes par tests-multiples,
2. les méthodes de pénalisation,
3. les méthodes par pénalisation des coefficients de régression,
4. les méthodes bayésiennes

A la fin de chaque paragraphe, nous résumons dans un encadré la méthode retenue, la raison pour laquelle nous avons retenu cette méthode et ses avantages et inconvénients.

Pour illustrer les différentes méthodes nous utilisons deux processus que nous avons simulés de façon à ce que la structure d'indépendance conditionnelle de ces processus soit le graphe de la figure 1.5. La méthode de simulation est expliquée dans le chapitre 2. La différence entre ces deux processus est qu'ils n'ont pas le même nombre d'observations $n$, $n \in \{60, 600\}$. Les performances des méthodes, notamment en fonction du nombre d'observations, sont discutées en détails dans le chapitre 4. Nous notons ces processus $\mathbf{X}^{(60)}$ et $\mathbf{X}^{(600)}$.




graphe

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

matrice d'adjacence

FIGURE *1.5 – Structure d'indépendance conditionnelle des processus* $\mathbf{X}^{(60)}$ *et* $\mathbf{X}^{(600)}$ *que nous allons utiliser pour illustrer le fonctionnement des méthodes.*

### 1.2.1 Méthodes par tests-multiples

Pour l'approche d'estimation de modèles graphiques gaussiens en estimant les arêtes via des tests multiples, le problème consiste à faire $\frac{p(p-1)}{2}$ tests soit un test par arête. Le test sur l'arête $(i, j)$ est le suivant :

$$\mathcal{H}_0(i,j) : \pi_{ij} = 0 \text{ et } \mathcal{H}_1(i,j) : \pi_{ij} \neq 0. \tag{1.4}$$

La première étape des procédures itératives présentées par Edwards [Edw00] est de tester ces deux hypothèses. Ensuite, deux approches peuvent être envisagées : une approche directe et une approche rétrograde.

- L'approche directe consiste à partir du graphe vide (sans arêtes) et de tester chaque arête pour savoir si elle peut être ajoutée au graphe en utilisant un test du $\chi^2$ basé sur la déviance entre deux modèles successifs. La déviance entre deux modèles est la différence entre les log vraisemblances de ces deux modèles : à l'itération $i$ dev $= n \log\left(|\Sigma_{i-1}^{emp}|/|\Sigma_i^{emp}|\right)$ (cf [Edw00], page 40). Les arêtes les plus significatives, c'est-à-dire celles avec les plus petites $p$-valeurs au test du $\chi^2$ sont ajoutées au graphe vide. La même procédure est répétée sur les arêtes non ajoutées aux itérations précédentes jusqu'à ne plus avoir aucune arête significative (c'est-à-dire que la plus petite $p$-valeur est plus grande que $\alpha$ où $\alpha$ est le risque de faux positif), la procédure est alors arrêtée.
- L'approche rétrograde consiste à supprimer des arêtes à partir du graphe complet : à chaque itération, les arêtes non-significatives, c'est-à-dire avec les plus importantes $p$-valeurs, sont supprimées du graphe courant. La procédure s'arrête quand toutes les arêtes sont significatives.

Pour ces approches, seules les erreurs sur les tests individuels sont contrôlées par $\alpha$ et non l'erreur sur l'ensemble du graphe $\alpha_{tot}$.

Drton et Perlman [DP04] proposent une approche qui fait le test (1.4) simultanément sur toutes les valeurs de corrélations partielles. Un rappel de cette méthode est rédigé dans l'annexe B. Cette approche est implémentée dans le paquet R **SIN** [DP08]. **SIN** signifie : Sure/Incertain/Nul. Ce paquet permet de construire deux types de graphes : celui contenant uniquement les arêtes qui sont très significatives $G_S$ et celui contenant aussi bien les arêtes très significatives que celles un peu moins significatives $G_{SI}$. Dans tous les cas, les graphes ne contiennent pas les arêtes non significatives.

Illustrons cette méthode sur les deux processus $\mathbf{X}^{(60)}$ et $\mathbf{X}^{(600)}$. Nous construisons $G_S$ en prenant les $p$-valeurs inférieures à $\alpha = 0.1$ et $G_{SI}$ en prenant les $p$-valeurs inférieures à $\alpha = 0.9$. Nous prenons ces valeurs de $\alpha$ car ce sont celles utilisées dans un exemple donné par Drton [DP08].

Pour le processus $\mathbf{X}^{(600)}$, le graphe $G_S$ et le graphe $G_{SI}$ sont les mêmes et ils correspondent à la structure d'indépendance conditionnelle du processus. Pour le processus $\mathbf{X}^{(60)}$, $G_S$ est inclus dans la structure d'indépendance conditionnelle du processus, ce qui est cohérent avec la définition de $G_S$ (c'est le graphe qui contient les arêtes qui sont assurément présentes dans la structure à estimer). Par définition, le graphe $G_{SI}$ est censé contenir la structure attendue car il est construit à partir des arêtes sûres et des arêtes incertaines. Or il ne contient pas la structure d'indépendance conditionnelle attendue, cela signifie qu'au moins une arête attendue est considérée comme significativement non présente dans le graphe. Ces observations mettent en avant que pour certaines valeurs de $n$, ici $n = 60$, les performances de la méthode SIN ne sont pas optimales. Ce point est discuté dans le chapitre 4.

La méthode SIN proposée par Drton et Perlman est basée sur une approche par tests multiples. Elle permet d'obtenir des informations sur la catégorie de chacune des arêtes : une arête est soit significative et est donc présente dans la structure d'indépendance conditionnelle du processus étudié, soit elle a une $p$-valeur très grande et n'est donc pas présente dans la structure, soit elle a une $p$-valeur qui ne permet pas de conclure si l'arête est dans la structure ou non. La force de cette méthode est de donner des informations sur la pertinence des arêtes plutôt que sur la structure en général. Cette méthode est également extrêmement rapide et donne d'excellents résultats si $n >> p$. La faiblesse est le choix des valeurs du risque de faux positif $\alpha$ pour séparer

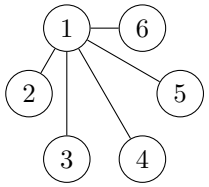
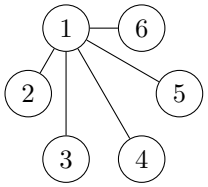
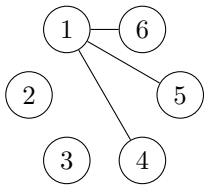
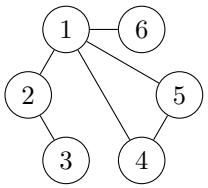
| | matrice des $p$-valeurs | $G_S$ | $G_{SI}$ |
|---|---|---|---|
| $\mathbf{X}^{(600)}$ | 0 0 0 0 0 0<br>0 0 0.999 0.999 0.999 0.999<br>0 0.999 0 0.999 0.914 0.999<br>0 0.999 0.999 0 0.999 0.999<br>0 0.999 0.914 0.999 0 0.996<br>0 0.999 0.999 0.999 0.996 0 |  |  |
| $\mathbf{X}^{(60)}$ | 0 0.106 0.994 0.001 0.035 0<br>0.106 0 0.793 0.994 0.994 0.994<br>0.994 0.793 0 1 0.994 0.994<br>0.001 0.994 1 0 0.437 0.994<br>0.035 0.994 0.994 0.437 0 0.950<br>0 0.994 0.994 0.994 0.950 0 |  |  |



FIGURE *1.6 – Résultats obtenus en utilisant la méthode SIN sur les processus* $\mathbf{X}^{(60)}$ *et* $\mathbf{X}^{(600)}$*. Les graphes* $G_S$ *et* $G_{SI}$ *sont obtenus en ne gardant que les arêtes associées aux p-valeurs inférieures respectivement à* $\alpha = 0.1$ *(en rouge) et* $\alpha = 0.9$ *(en rouge et vert). Pour le processus* $\mathbf{X}^{(600)}$*,* $G_S$ *et* $G_{SI}$ *correspondent à la structure d'indépendance conditionnelle attendue. Pour* $\mathbf{X}^{(60)}$*,* $G_S$ *est inclus dans la structure d'indépendance conditionnelle attendue mais* $G_{SI}$ *n'est pas inclus dans la structure attendue et il ne contient pas non plus cette structure.*

les trois catégories d'arêtes. Par exemple, dans le cas du processus $\mathbf{X}^{(60)}$, si nous avions choisi $\alpha = 0.11$, l'arête $(1, 2)$ aurait appartenu au graphe $G_S$ (voir Annexe B).

**Méthode SIN**
*Raison :* méthode de tests-multiples accessible sous R.
✓*Avantages :*
- méthode rapide
- donne une information sur la pertinence statistique de chaque arête

✗*Inconvénients :*
- les graphes solutions dépendent du choix du risque de faux positif $\alpha$

Information Toolbox : Dans notre Toolbox, nous avons créé une fonction Matlab qui exécute la fonction SIN adaptée aux graphes non-dirigés en R. Cette fonction s'appelle *Method_ sinUG_ R*, elle prend en entrée un processus multivarié gaussien et retourne les graphes $G_S$ et $G_{SI}$.
Le paquet SIN est disponible à l'adresse suivante : http://cran.r-project.org/web/packages/SIN/

### 1.2.2 Méthodes par pénalisation de la matrice de précision

La majorité des méthodes de pénalisation se base sur la méthode du *lasso* développée par Tibshirani [Tib96]. Cette méthode est utilisée pour faire de la régression linéaire, elle introduit un terme qui pénalise selon la norme $\ell_1$ les coefficients de régression afin de favoriser les solutions parcimonieuses, c'est-à-dire qu'un nombre important de coefficients de régression estimés sont nuls. Concernant l'estimation des modèles graphiques gaussiens, il existe de nombreuses méthodes qui ont été développées à partir du *lasso* au cours de la dernière décennie. En plus des méthodes qui pénalisent selon la norme $\ell_1$, Tibshirani [Tib11] donne un échantillon de méthodes qui pénalisent selon d'autres normes, méthodes auxquelles nous ne nous sommes pas intéressés.

#### 1.2.2.1 Le Graphical lasso

Nous détaillons davantage cette méthode car elle est utilisée dans la méthode ABiGlasso que nous avons développée et que nous présentons dans le chapitre 3.

La méthode la plus connue basée sur le *lasso* est le Graphical lasso [FHT08] : la log-vraisemblance est pénalisée en $\ell_1$ afin d'obtenir une estimée de la matrice de précision parcimonieuse :

$$\widehat{K} = \underset{K}{\operatorname{argmax}} \log(\det(K)) - \operatorname{tr}(K\Sigma^{emp}) - \lambda||K||_1 \tag{1.5}$$

où $\lambda$ est le paramètre de pénalisation et $\Sigma^{emp}$ la matrice de covariance empirique.
Le taux de parcimonie, c'est-à-dire le nombre de valeurs nulles de la solution du Graphical lasso est imposé par le paramètre de pénalisation : si celui-ci est proche de 0, la solution du Graphical lasso sera peu parcimonieuse, et plus $\lambda$ augmente, plus la matrice solution est parcimonieuse jusqu'à être égale à la matrice nulle. Le graphe $\widehat{G}_\lambda$ est construit de la façon suivante : $(i,j) \in E$ si $\hat{k}_{ij} \neq 0$ et $(i,j) \notin E$ si $\hat{k}_{ij} = 0$ avec $\widehat{K} = [\hat{k}_{ij}]$.

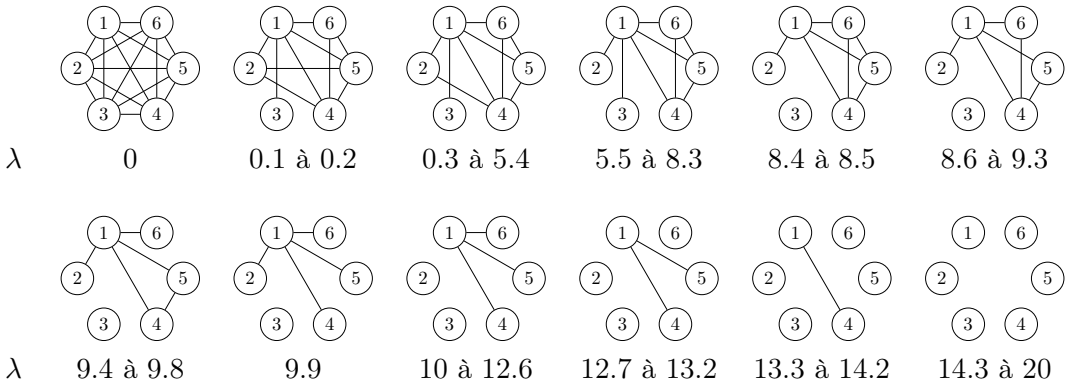



FIGURE *1.7 – Évolution du graphe solution du Graphical lasso en fonction du choix de $\lambda$ pour le processus $\mathbf{X}^{(600)}$. Pour $\lambda = 0$ le graphe est complet et au fur et à mesure que $\lambda$ augmente, le graphe solution a de moins en moins d'arêtes jusqu'à avoir un graphe vide ($E = \emptyset$) à partir de $\lambda = 14.3$.*

La figure 1.7 illustre le fait que la solution obtenue dépend de $\lambda$. Pour $\lambda = 0$ le graphe est complet et à partir d'une certaine valeur de $\lambda$ (ici 14.3) le graphe n'a plus d'arête.

Concernant cette méthode, deux problèmes se présentent :

1. le réglage de $\lambda$,
2. la possible non convergence de l'algorithme

**Réglage de $\lambda$**

Le problème du réglage de $\lambda$ est souvent résolu en utilisant la validation croisée à $k$ replis (fold) : pour un processus à $n$ observations, les données sont découpées de façon à avoir un processus à $(k-1)/k \times n$ observations et un deuxième à $1/k \times n$ observations. Le Graphical lasso est appliqué sur le processus à $(k-1)/k \times n$ observations pour obtenir une estimée de la matrice de précision par valeur de $\lambda$ : $\widehat{K}_\lambda$. Le $\lambda$ retenu est celui associé à la matrice $\widehat{K}_\lambda$ qui maximise la log-vraisemblance avec comme matrice de covariance empirique celle du processus à $1/k \times n$ observations. Par exemple, Friedman *et al.* [FHT08] font de la validation croisée à 10 replis. Cette méthode nécessite cependant d'avoir un nombre d'observations grand par rapport au nombre de variables pour donner des résultats pertinents.

**Exemple de non consistance de l'estimateur du Graphical lasso**

Meinshausen [Mei08] illustre sur un exemple le fait que la solution proposée par le Graphical lasso peut ne pas être cohérente avec les données. Il choisit $G = (V, E)$ avec $V = \{1, 2, 3, 4\}$ et $E = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 4)\}$), il définit une matrice de covariance $\Sigma$ de la façon suivante :

$$\begin{cases} \sigma_{ii} = 1 & \text{pour } i \in V \\ \sigma_{ij} = \rho & \text{pour } (i, j) \in E \\ \sigma_{ij} = 0 & \text{pour } (i, j) \notin E \end{cases}$$

avec $\rho \in [0, 1/\sqrt{2}]$. Il prouve que si $\rho > (3/2)^{1/2} - 1 \simeq 0.23$ alors quand $n \to \infty$, $E$ est mal estimé avec une probabilité supérieure à 0.5. Ravikumar *et al.* [RWRY11] donnent un autre exemple de non consistance de l'estimation par Graphical lasso sur un graphe en étoile : $G = (V, E)$ avec $V = \{1, 2, 3, 4\}$ et $E = \{(1, 2), (1, 3), (1, 4)\}$. $\Sigma$ est défini de la façon suivante :

$$\begin{cases} \sigma_{ii} = 1 & \text{pour } i \in V \\ \sigma_{ij} = \rho & \text{pour } (i, j) \in E \\ \sigma_{ij} = \rho^2 & \text{pour } (i, j) \notin E \end{cases}$$

L'estimateur n'est pas consistant si $|\rho| \in [0.4, 1]$.

L'algorithme de résolution du Graphical lasso (1.5) proposé par Friedman *et al.* [FHT08], basé sur une approche par descente de coordonnées rapide, n'est pas la seule méthode d'optimisation, d'autres algorithmes sont proposés mais les problèmes rencontrés sont les mêmes [BGd08, BPC+10]. Le point fort de ces méthodes est cependant leur rapidité d'exécution. Dans le cadre de nos travaux, nous utilisons l'algorithme de Boyd [BPC+10] basé sur la méthode de multiplicateurs à directions alternées.

**Méthode Graphical lasso**
*Raison :* méthode très utilisée.
✓*Avantages :* méthode rapide
✗*Inconvénients :*
- dépend du choix du paramètre de pénalisation $\lambda$
- peut ne pas converger

Information Toolbox : la version de Boyd [BPC+10] du Graphical lasso est disponible dans notre Toolbox, c'est la fonction *covsel*. Elle prend en entrée un processus multivarié gaussien et une valeur du paramètre de pénalisation et donne une matrice de précision parcimonieuse. Cette fonction est disponible en ligne à l'adresse suivante : http://stanford.edu/~boyd/papers/admm/.

#### 1.2.2.2 Interprétation bayésienne du Graphical lasso

Wang [Wan12] propose une interprétation bayésienne du Graphical lasso en exprimant la loi de probabilité a posteriori de la matrice de précision $K$ sachant $\lambda$. Plus de détails sur cette méthode sont donnés dans l'annexe C.

Cette méthode a pour solution une matrice de précision $\widehat{K}_{BAGlasso}$ en général non parcimonieuse. Si l'objectif est d'obtenir une structure de graphe, Wang propose d'utiliser un "seuillage

bayésien". Ce seuillage se fait sur la matrice des corrélations partielles $\widehat{\Pi}_{BAGlasso}$ qui est la version normalisée de la matrice de précision $\widehat{K}_{BAGlasso}$. La condition de seuillage est formulée ainsi :

$$\left|\frac{(\widehat{\Pi}_{BAGlasso})_{ij}}{\mathbb{E}(\Pi_{ij}|\mathbf{X})}\right| > 0.5 \tag{1.6}$$

où $\mathbb{E}(\Pi_{ij}|\mathbf{X})$ est, dans [Wan12], la moyenne sur 1000 tirages de la simulation de la matrice des corrélations partielles $\Pi$ : on génère $K$ selon $\mathcal{W}(3+n, I_p + \Sigma^{emp})$ qui est une loi de Wishart (cf 1.1.2) avec $3+n$ degrés de liberté et pour matrice d'échelle la somme de la matrice de covariance empirique $\Sigma^{emp}$ et $I_p$ la matrice identité de taille $p \times p$. Ensuite on normalise pour avoir $\Pi$. Ce seuillage se base sur le travail de Carvalho *et al.* [CPJ10].

Par exemple, pour le processus $\mathbf{X}^{(600)}$, la matrice $\widehat{\Pi}$ obtenue n'est pas parcimonieuse mais a de très faibles valeurs en valeur absolue (cf figure 1.8). En seuillant cette matrice à l'aide du "seuillage bayésien", le graphe obtenu correspond à la structure de graphe attendue.

$\widehat{\Pi}_{BAGlasso}$

| | | | | | |
|---|---|---|---|---|---|
| 1 | -0,28 | -0,21 | 0,61 | -0,54 | -0,46 |
| -0,28 | 1 | $-1.48{\cdot}10^{-3}$ | $-6{,}60\cdot10^{-7}$ | $2{,}39\cdot10^{-4}$ | $-4{,}12{\cdot}10^{-5}$ |
| -0,21 | $-1.48{\cdot}10^{-3}$ | 1 | $-1.54{\cdot}10^{-3}$ | $-5{,}98{\cdot}10^{-6}$ | $1.84{\cdot}10^{-4}$ |
| 0,61 | $-6{,}60{\cdot}10^{-7}$ | $-1.54{\cdot}10^{-3}$ | 1 | $2.07{\cdot}10^{-3}$ | $-3{,}42{\cdot}10^{-3}$ |
| -0,54 | $2{,}39{\cdot}10^{-4}$ | $-5{,}98{\cdot}10^{-6}$ | $2.07{\cdot}10^{-3}$ | 1 | $-4.38{\cdot}10^{-4}$ |
| -0,46 | $-4{,}12{\cdot}10^{-5}$ | $1.84{\cdot}10^{-4}$ | $-3{,}42{\cdot}10^{-3}$ | $-4.38{\cdot}10^{-4}$ | 1 |

$\widehat{G}$

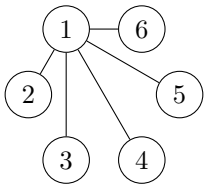



FIGURE *1.8 – Matrice des corrélations partielles estimée à partir de la méthode BAGlasso pour le processus* $\mathbf{X}^{(600)}$ *et graphe obtenu après application du seuillage bayésien de cette dernière.*

Cette approche a pour avantage de s'affranchir du choix du paramètre de pénalisation $\lambda$. Cependant la solution obtenue est non parcimonieuse et nécessite l'utilisation d'un seuillage qui dépend des paramètres de la fonction de Wishart utilisée pour effectuer ce seuillage.

**Méthode BAGlasso**
*Raison :*
- approche à base du Graphical lasso estimant le paramètre de pénalisation $\lambda$
- codes disponibles sur le site de l'auteur.

✓*Avantages :* il n'est plus nécessaire de choisir $\lambda$.
✗*Inconvénients :* construction du graphe par seuillage de la matrice des corrélations partielles, seuillage dépendant des paramètres.

Information Toolbox : la méthode BAGlasso est présente dans notre Toolbox. La fonction s'appelle *Method_BAGlasso*, elle prend en entrée la matrice de covariance empirique et le nombre d'observations du processus étudié et donne la matrice de précision estimée par la méthode et le graphe après "seuillage bayésien".
La méthode est implémentée par Wang et disponible à l'adresse suivante : https://www.msu.edu/~haowang/RESEARCH/Bglasso/bglasso.html
Nous avons implémenté le seuillage bayésien d'après l'article [Wan12].

### 1.2.3 Méthodes par pénalisation des coefficients de régression

Giraud *et al.* [Gir08][GHV12] proposent d'estimer la structure d'indépendance conditionnelle d'un processus multivarié gaussien en estimant les coefficients de régression linéaire nuls. Pour

ce faire, ils introduisent un critère qui pénalise les coefficients de régression linéaire en fonction d'un graphe $G$, ce critère ayant pour objectif de minimiser l'erreur quadratique de prédiction. Le graphe qui minimise ce critère est choisi comme représentant de la structure d'indépendance conditionnelle du processus. Le temps de calcul associé à une étude exhaustive sur l'ensemble des graphes possibles peut s'avérer rédhibitoire. Afin de réduire le temps de calcul, ils proposent de se limiter à l'étude de familles de graphes ayant certaines propriétés. La famille C01 est obtenue par tests statistiques sur les matrices de corrélation et des corrélations partielles, la famille LA par pénalisation $\ell_1$ des coefficients de régression, la famille EW est similaire à la famille LA mais sa pénalisation utilise un estimateur des poids exponentiels et la famille QE qui réduit le problème de la minimisation du critère à $p$ problèmes (au lieu de $p(p-1)/2$ problèmes).

Plus de détails sur le critère et les familles de graphes proposées sont donnés dans l'annexe D.

Nous utilisons la méthode GGMselect avec en paramètre les trois familles C01, LA et EW pour estimer la structure d'indépendance conditionnelle de $\mathbf{X}^{(60)}$. La figure 1.9 présente les graphes obtenus pour chacune des trois familles avec leur critère (cf (D.1) annexe D).

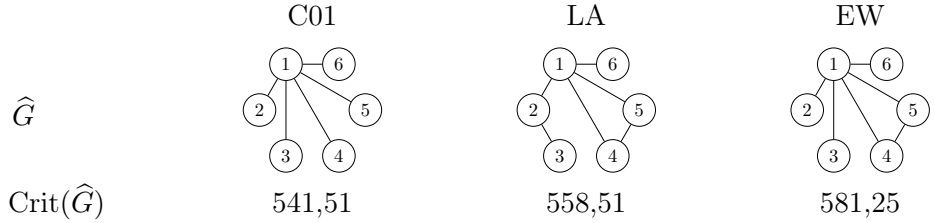



FIGURE *1.9 – Résultats de la méthode GGMselect sur le processus* $\mathbf{X}^{(60)}$ *pour les familles C01, LA et EW.*

Les trois familles donnent un graphe solution différent. Cependant, en utilisant le critère associé à chacun des graphes, c'est le graphe obtenu avec la famille C01 qui est le plus probable car c'est celui qui minimise le critère. Ce graphe correspond en effet à la structure d'indépendance conditionnelle attendue. Pour le processus $\mathbf{X}^{(600)}$, le graphe obtenu est le même pour toutes les familles et correspond au graphe attendu.

Ce qui a retenu notre attention dans la méthode de Giraud est la volonté de réduire l'ensemble des graphes à parcourir pour limiter le temps de calcul. De plus, l'utilisation du critère pour comparer les résultats de différentes méthodes est un atout qui n'est pas fourni par beaucoup d'autres méthodes.

Notons que le choix de certains paramètres pour le calcul du critère ainsi que le choix de la famille à utiliser reste à l'appréciation de l'utilisateur.

**Méthode GGMselect**
*Raison :* seule méthode avec approche sur les coefficients de régression, disponible sous R.
✓*Avantages :*
- critère permettant de comparer différentes solutions
- réduction du nombres de graphes à parcourir en travaillant sur des familles de graphes

✗*Inconvénients :*
- choix de la famille et de certains paramètres arbitraires
- codes ne fournissant que le critère du graphe solution

Information Toolbox : Dans notre Toolbox, nous avons créé une fonction Matlab qui exécute la fonction GGMselect associée à la famille choisie en R. Cette fonction s'appelle *Method_ GGMselect_ R*, elle prend en entrée un processus multivarié gaussien et une famille au choix parmi celles présentées dans cette section et donne le graphe solution et la valeur de son critère.
Le paquet GGMselect est disponible à l'adresse suivante : http://cran.r-project.org/web/packages/GGMselect/.

### 1.2.4 Méthodes bayésiennes

Dans cette partie, nous étudions diverses méthodes d'estimation de modèles graphiques gaussiens. Ces méthodes ont pour point commun d'utiliser une approche bayésienne. Afin de comprendre ces méthodes, nous faisons quelques rappels sur le principe d'une approche bayésienne.

#### 1.2.4.1 Principe des méthodes bayésiennes

Soit $\mathbf{Y}$ un vecteur d'observations, $\mathbf{Y} \sim f(\mathbf{Y}|\boldsymbol{\theta})$ où $f$ est une fonction connue mais dont le paramètre $\boldsymbol{\theta} \in \Theta$ est inconnu. L'objectif d'une approche bayésienne est d'exploiter l'information apportée par $\mathbf{Y}$ pour estimer $\boldsymbol{\theta}$.

**Définition 1.2.1.** ***Vraisemblance***
*La vraisemblance $\ell(\boldsymbol{\theta})$ est une fonction exprimant que $\boldsymbol{\theta}$ dépend des observations $\mathbf{Y}$. $\ell(\boldsymbol{\theta}|\mathbf{Y}) = f(\mathbf{Y}|\boldsymbol{\theta})$ car l'information fournie par $\mathbf{Y}$ est contenue dans $f(\mathbf{Y}|\boldsymbol{\theta})$.*

**Définition 1.2.2.** ***Distribution a priori***
*L'incertitude sur le paramètre $\boldsymbol{\theta}$ peut être décrite par une distribution de probabilité $\pi$ appelée distribution a priori. Cela revient à supposer que $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$ sans connaître les observations $\mathbf{Y}$.*

**Définition 1.2.3.** ***Distribution a posteriori***
*$\mathbb{P}(\boldsymbol{\theta}|\mathbf{Y})$ est la distribution de probabilité de $\boldsymbol{\theta}$ connaissant les observations $\mathbf{Y}$. Cette distribution est appelée distribution a posteriori et est définie de la façon suivante :*

$$\mathbb{P}(\boldsymbol{\theta}|\mathbf{Y}) = \frac{\ell(\boldsymbol{\theta}|\mathbf{Y})\pi(\boldsymbol{\theta})}{\int_\Theta \ell(\boldsymbol{\theta}|\mathbf{Y})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}}.$$

Dans le cas des modèles graphiques gaussiens, $\boldsymbol{\theta}$ est le graphe $G$ représentant la structure d'indépendance conditionnelle et $\mathbf{Y}$ sont nos processus $\mathbf{X}$. Comme la marginale ne dépend pas du graphe, la distribution a posteriori peut être écrite de la façon suivante :

$$\mathbb{P}(G|\mathbf{X}) \propto \pi(G)\ell(G|\mathbf{X}) = \pi(G)f(\mathbf{X}|G) \tag{1.7}$$

#### 1.2.4.2 Sur les graphes décomposables

Nous avons vu dans la section 1.1.3.2 que dans le cas de graphes décomposables, la distribution de $\mathbf{X}$ se factorise de la façon suivante si elle est strictement positive :

$$\mathbb{P}(\mathbf{X}) = \frac{\prod_{C\in\mathcal{C}} \mathbb{P}(X_C)}{\prod_{S\in\mathcal{S}} \mathbb{P}(X_S)} \tag{1.8}$$

Une distribution gaussienne étant strictement positive, cette propriété permet de calculer aisément la densité de probabilité a posteriori du graphe $G$ sachant le processus gaussien $\mathbf{X}$ mais uniquement dans le cas où le graphe qui lui est associé est décomposable :

$$\mathbb{P}(G|\mathbf{X},\delta,\Phi) \propto \frac{h_G(\delta,\Phi)}{h_G\big(\delta+n,\Phi+nS(\mathbf{X})\big)}\pi(G) \tag{1.9}$$

avec :

$$h_G(\delta,\Phi) = \frac{\prod_{C\in\mathcal{C}} \frac{\det\left(\frac{\Phi_C}{2}\right)^{(\delta+|V_C|-1)/2}}{\Gamma_{|V_C|}\left(\frac{\delta+|V_C|-1}{2}\right)}}{\prod_{S\in\mathcal{S}} \frac{\det\left(\frac{\Phi_S}{2}\right)^{(\delta+|V_S|-1)/2}}{\Gamma_{|V_S|}\left(\frac{\delta+|V_S|-1}{2}\right)}}$$

Le détail du calcul est donné dans l'annexe E

La probabilité a posteriori d'un graphe sachant les données est connue (équation (1.9)). Cependant, elle dépend du choix de la distribution a priori sur le graphe $G$ et des paramètres $\delta$ et $\Phi$ de la fonction hyper-inverse Wishart (loi a priori $\pi(\Sigma|G)$ pour les graphes décomposables selon Dawid et Lauritzen [DL93]). Les méthodes bayésiennes d'estimation de structures d'indépendance conditionnelle avec solution dans l'espace des graphes décomposables diffèrent sur le choix de ces trois éléments ainsi que sur la manière de parcourir les graphes.

**Modèles et méthodes utilisés**

Pour le choix de $\pi(G)$, deux a priori sont utilisés : la loi uniforme qui ne met aucun a priori sur la structure du graphe et la loi binomiale de paramètre $r$ avec $r$ choisi de façon à privilégier les graphes avec peu ou beaucoup d'arêtes (les graphes ayant le même nombre d'arêtes sont équiprobables). Cela dépend du modèle choisi par les auteurs.

Pour le choix des paramètres $\delta$ et $\Phi$, là encore cela dépend du modèle choisi par les auteurs. Dans l'annexe F, section F.1, un inventaire des choix faits par différents auteurs est réalisé.

Il n'existe pas de solution meilleure qu'une autre, ce sont différents modèles envisagés et ayant leurs propres motivations et justifications.

Les méthodes varient aussi en fonction de l'approche utilisée pour parcourir l'ensemble des graphes. Plusieurs exemples de méthodes de parcours de l'ensemble des graphes sont donnés dans la section F.2 de l'annexe F. Nous retiendrons qu'il est coûteux en temps de calcul de parcourir la totalité des graphes et que les différentes méthodes ont pour objectif de proposer une approche permettant de réduire le temps de calcul sans nuire à la qualité de la solution proposée.

**Taille de l'espace des graphes décomposables par rapport à l'ensemble des graphes**

Il n'existe pas de méthode pour dénombrer le nombre de graphes décomposables pour un nombre de nœuds $p$ donné. Cependant, la proportion de graphes décomposables par rapport à tous les graphes possibles décroît comme $p$ augmente. Giudici et Green [Giu96] donnent des exemples :

- pour $p = 4$, il y a 61 graphes décomposables sur 64 graphes possibles,
- pour $p = 6$, il y a 80% de graphes décomposables dans l'ensemble des graphes possibles,
- pour $p = 16$, il n'y a plus que 45% de graphes décomposables.

Puisque la proportion de graphes décomposables dans l'ensemble des graphes possibles diminue comme $p$ augmente, il ne semble pas pertinent de se focaliser uniquement sur les graphes décomposables quand aucun a priori n'est connu sur le graphe recherché.

Les méthodes bayésiennes sur les graphes décomposables tirent bénéfice de la possibilité d'exprimer exactement la vraisemblance. Cependant ces méthodes imposent que la solution obtenue soit décomposable, les graphes décomposables pouvant n'être qu'un petit sous-ensemble des graphes possibles.

Le choix de travailler avec des graphes décomposables est aussi motivé par le fait que le nombre de graphes à parcourir est moindre que si il fallait explorer l'ensemble des graphes (décomposables et non-décomposables) et donc que le temps de calcul nécessaire pour parcourir cet ensemble est plus abordable.

**Méthode retenue**

Pour la suite, nous prenons comme représentante des méthodes bayésiennes sur les graphes décomposables la méthode FINCS de Scott et Carvalho [SC08] car elle permet d'obtenir un résultat dans un temps raisonnable pour les petits graphes. Cette méthode donne également un score sur les arêtes permettant d'évaluer la pertinence de la présence des différentes arêtes dans le graphe solution.

Pour le processus test $\mathbf{X}^{(600)}$, la figure 1.10 donne les 5 graphes les plus probables et leur log-vraisemblance. Le figure 1.11 donne les scores d'inclusion des arêtes.

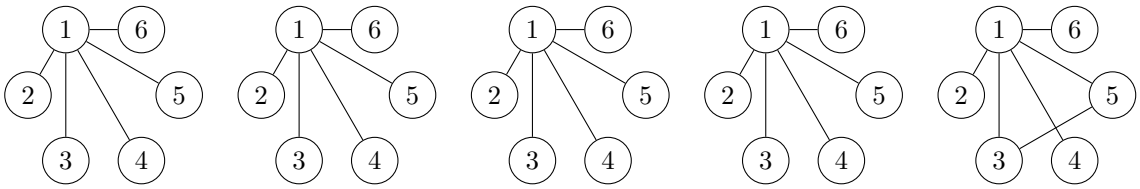



FIGURE *1.10 – Les cinq graphes les plus probables donnés par la méthode FINCS pour représenter la structure d'indépendance conditionnelle du processus* $\mathbf{X}^{(600)}$ *par ordre de probabilité décroissante. Chaque graphe est donné avec sa log-vraisemblance.*

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0.021 | 0.021 | 0.021 | 0.022 |
| 1 | 0.021 | 0 | 0.022 | 0.042 | 0.021 |
| 1 | 0.021 | 0.022 | 0 | 0.020 | 0.021 |
| 1 | 0.021 | 0.042 | 0.020 | 0 | 0.026 |
| 1 | 0.022 | 0.021 | 0.021 | 0.026 | 0 |

FIGURE *1.11 – Scores d'inclusion pour chaque arête de la structure d'indépendance conditionnelle du processus* $\mathbf{X}^{(600)}$*. Ces scores sont obtenus en utilisant la méthode FINCS.*

Nous constatons que les quatre graphes ayant les log-vraisemblances les plus élevées sont en fait le même graphe. Ce phénomène provient du fait qu'un même graphe peut être parcouru plusieurs fois et que la log vraisemblance est recalculée à chaque fois en prenant en compte les paramètres mis à jour lors des précédentes itérations. Ainsi, un même graphe n'a pas la même log-vraisemblance en fonction de l'itération à laquelle il est parcouru.

**Méthode FINCS**

*Raison :* méthode parmi les plus rapides pour une approche bayésienne sur les graphes décomposables

✓*Avantages :* permet de comparer les graphes explorés et donc de juger de la pertinence de la solution proposée

✗*Inconvénients :*

- se limite aux graphes décomposables
- temps de calcul long

Information Toolbox : Aucune méthode bayésienne sur les graphes décomposables n'a été introduite dans notre Toolbox pour le moment à cause de soucis de compatibilité.
Cependant, la méthode FINCS est disponible à l'adresse suivante : http://www.tandfonline.com/doi/suppl/10.1198/106186008X382683#tabModule.

#### 1.2.4.3 Sur les graphes non-décomposables

Quand l'ensemble des graphes parcouru est l'ensemble des graphes décomposables, les composantes primaires (définition 1.1.20) sont toutes complètes et la matrice de covariance suit une loi inverse Wishart. Si l'ensemble des graphes parcouru n'est pas contraint, la loi suivie par la matrice de covariance doit être estimée.

Bien que pour les graphes non-décomposables, la propriété de factorisation (1.8) ne s'applique pas aussi facilement que pour les graphes décomposables, elle est à la base des travaux de Roverato [Rov02], de Dellaportas *et al.* [DGR03] et de Atay-Kayis et Massam [AKM05] qui ont pour objectif de calculer la probabilité a posteriori dans le cas de graphes non-décomposables.

Marrelec [MB06a], en se basant sur les travaux de Roverato [RW98, Rov99, Rov02], a proposé une formulation de la probabilité a posteriori valable pour tout graphe (décomposable ou non). Cette formulation est asymptotique, c'est-à-dire valable quand le nombre d'observations tend vers l'infini. Nous notons $\stackrel{a}{=}$ lorsque l'égalité est atteinte asymptotiquement :

$$\mathbb{P}(G|\mathbf{X}) \stackrel{a}{=} \frac{1}{C(\mathbf{X})} \times \frac{\varphi_{\boldsymbol{\pi}_{\bar{E}}^{emp}, cW_{\bar{E}\bar{E}}}(\mathbf{0})}{V(G)} \tag{1.10}$$

où $C(\mathbf{X})$ est une constante dépendant uniquement des données et non de $G$. $\varphi_{\mu,\Sigma}(\mathbf{Y})$ est la loi normale de moyenne $\mu$, de covariance $\Sigma$ appliquée au vecteur $\mathbf{Y}$. La constante $c = 1/(n+p+1)$, $W$ est la matrice d'Isserlis [RW98] de la matrice de précision empirique, $\boldsymbol{\pi}$ est le vecteur des corrélations partielles selon le support de $G$, les indices $\bar{E}$ et $\bar{E}\bar{E}$ indiquent que seules les valeurs associées aux arêtes absentes de $G$ sont considérées : $\bar{E}$ est le complémentaire de $E$ sur l'ensemble des arêtes possibles. $W_{\bar{E}\bar{E}}$ est la matrice $W$ dont seules les lignes et les colonnes correspondant à $\bar{E}$ ont été conservées. $V(G)$ est le volume des matrices de corrélations partielles définies positives ayant des valeurs nulles là où le graphe $G$ n'a pas d'arêtes. Le terme $\varphi_{\boldsymbol{\pi}_{\bar{E}}^{emp}, cW_{\bar{E}\bar{E}}}(\mathbf{0})$ quantifie à quel point les valeurs de corrélations partielles empiriques associées aux arêtes absentes sont proches de 0 qui est la valeur qu'elles devraient avoir en théorie.

Le détail du calcul de cette probabilité est donné dans l'annexe G.

Cette probabilité a posteriori peut être utilisée pour une recherche exhaustive, ou pour une méthode bayésienne à l'aide d'un échantillonneur de Gibbs comme présenté dans [MB06a]. Cependant utiliser cette probabilité présente un inconvénient majeur dans le cas où on souhaite explorer beaucoup de graphes : l'estimation du volume $V(G)$ est très coûteuse en temps de calcul ($1/10^e$ de seconde par graphe pour des graphes à 6 nœuds).

Nous utilisons les travaux de Marrelec pour développer une nouvelle méthode présentée dans le chapitre 3.

L'exploration des graphes décomposables est souvent préférée à l'exploration de tous les graphes. En effet l'ensemble des graphes décomposables est beaucoup plus petit que l'ensemble de tous les graphes, permettant ainsi d'avoir des temps d'exploration plus faibles. Cependant, travailler avec l'ensemble de tous les graphes possibles permet de ne poser aucun a priori sur le graphe attendu.

**Méthode de Marrelec**

*Raison :* fournit une formulation a posteriori de la probabilité d'un graphe dans le cadre général

✓*Avantages :*

- permet de comparer tous les graphes explorés
- n'impose aucun a priori sur les graphes à explorer

✗*Inconvénients :* temps de calcul très long voire rédhibitoire si on souhaite explorer l'ensemble des graphes possibles

## 1.3 Discussion

La définition d'un modèle graphique gaussien permet de représenter la structure d'indépendance conditionnelle d'un processus gaussien.

Nous avons vu dans cette section qu'il existe différentes approches pour estimer cette structure. Nous avons mis l'accent sur certaines méthodes dont nous résumons les avantages et les inconvénients dans le tableau 1.1.

| méthode | avantages | inconvénients |
|---|---|---|
| Tests multiples - SIN | ✓informations sur les arêtes<br>✓méthode très rapide | ✗seuils arbitraires pour séparer les groupes d'arêtes |
| Graphical lasso | ✓méthode rapide | ✗choix du paramètre de pénalisation difficile<br>✗possibilité de non convergence |
| Bayesian Adaptive Glasso | ✓permet de ne pas avoir à choisir le paramètre de pénalisation $\lambda$ | ✗seuillage avec paramètres à choisir pour avoir un graphe |
| Pénalisation des coefficients de régression - GGMselect | ✓réduction de l'ensemble des graphes à explorer à l'aide de familles de graphes<br>✓critère pour comparer les résultats de différentes méthodes | ✗accès uniquement au critère du graphe le plus probable<br>✗choix de la famille arbitraire |
| Approche bayésienne sur graphes décomposables - FINCS | ✓possibilité de comparer les graphes explorés | ✗graphes avec une topologie particulière<br>✗temps de calcul long |
| Approche bayésienne sur tous les graphes - Marrelec | ✓possibilité de comparer tous les graphes explorés<br>✓pas de contrainte sur la topologie des graphes explorés | ✗temps de calcul très long |

TABLEAU *1.1 – Tableau récapitulatif des avantages et des inconvénients des méthodes d'estimation des modèles graphiques gaussiens.*

Nous souhaitons mettre au point une méthode qui évite les deux problèmes suivants : être dépendante d'un paramètre difficile à choisir (comme les méthodes Graphical lasso ou SIN) et avoir une méthode trop onéreuse en temps de calcul comme les approches bayésiennes. Pour réduire le temps de calcul, nous adoptons une approche similaire à celle de Giraud *et al.*, c'est-à-dire que nous réduisons l'ensemble des graphes à explorer à un sous-ensemble de graphes ayant

certaines propriétés, nous utilisons le Graphical lasso pour construire cet ensemble. Concernant les méthodes bayésiennes, leur point fort est de donner la probabilité a posteriori du graphe sachant le processus. Cette probabilité permet de juger de la pertinence de la solution obtenue et nous souhaitons conserver cet aspect. Afin de n'imposer aucun a priori sur la structure du graphe solution, nous utilisons les travaux menés par Marrelec sur l'ensemble des graphes. Ces attentes et observations nous ont conduit à développer la méthode présentée dans le chapitre 3.

Nous avons également constaté qu'il était difficile de comparer les performances des différentes approches car elles sont souvent testées sur des processus réels dont la structure d'indépendance conditionnelle n'est pas connue ou sur un processus simulé qui varie d'une méthode à l'autre. Donc avant d'introduire une nouvelle méthode, nous introduisons les outils que nous allons utiliser pour évaluer cette méthode et pour la comparer aux méthodes déjà existantes. C'est le rôle du prochain chapitre.

En plus de comparer les performances des différentes méthodes, nous voulons également étudier l'impact du nombre d'observations sur la qualité de la solution proposée pour les différentes méthodes présentées. En effet, l'objectif est d'avoir une méthode donnant de bonnes performances dans le cas $n > p$, c'est-à-dire où le nombre d'observations d'un processus est grand devant le nombre de variables mais pas trop. En effet, dans la réalité il est peu probable d'avoir des processus avec un nombre d'observations très grand devant le nombre de variables. Le fait de travailler avec des processus gaussiens et donc de ne s'intéresser qu'aux moments d'ordre deux simplifie déjà le problème mais il n'en reste pas moins un problème complexe. Cette étude est menée dans le chapitre 4

Les exemples présentés dans cette partie sont construits à partir de processus avec $p = 6$ variables. Ce faible nombre de variables permet d'obtenir des résultats quelle que soit la méthode considérée. En effet, plus le nombre de variables augmente, plus le temps de calcul de certaines méthodes augmente jusqu'à devenir rédhibitoire. Pour réduire l'impact du temps de calcul, nous conduisons l'ensemble de nos études sur des processus à six variables, le passage à l'échelle est introduit à la fin du chapitre 4 afin d'illustrer les problèmes rencontrés par les différentes méthodes lorsque $p$ est grand ($p = 100$ pour le cas que nous avons étudié).

# Chapitre 2

# Procédure d'évaluation de méthodes d'estimation de modèles graphiques gaussiens

**Sommaire**



L'objectif des méthodes d'estimation de modèles graphiques gaussiens est de fournir le graphe le plus représentatif de la structure d'indépendance conditionnelle d'un processus multivarié gaussien. Pour évaluer les performances de ces méthodes il faut connaître la structure d'indépendance des processus de test et avoir au moins une métrique permettant de mesurer la distance entre le graphe proposé et la structure attendue. L'objet de ce chapitre est de proposer une procédure d'évaluation de méthodes d'estimation de modèles graphiques gaussiens prenant en compte ces différents aspects.

Ce chapitre est divisé en trois parties : dans la première partie nous nous concentrons sur la simulation de processus dont la structure d'indépendance conditionnelle est connue. Dans un premier temps nous présentons des exemples de processus utilisés dans la littérature, ensuite nous nous intéressons à des méthodes de simulations de processus et finalement nous introduisons notre méthode de simulation qui remédie à des défauts communs aux méthodes déjà existantes. Dans la seconde partie, nous nous intéressons aux métriques utilisées pour mesurer la distance entre deux graphes et nous proposons une métrique basique qui complète les métriques existantes. Dans la troisième partie, nous présentons une procédure de test pour évaluer les performances des méthodes d'estimation des modèles graphiques gaussiens.

Ce chapitre est motivé par le fait que trop peu d'articles proposant des méthodes d'estimation de modèles graphiques gaussiens présentent les performances de leurs méthodes sur des données simulées, c'est-à-dire dont la structure d'indépendance conditionnelle est connue.

## 2.1 Processus tests et simulations

Dans le cadre de cette thèse, nous souhaitons extraire la structure d'indépendance conditionnelle de processus multivariés gaussiens. Pour évaluer les performances aussi bien des méthodes existantes que des méthodes que nous allons proposer, nous souhaitons générer des processus pour lesquels la structure d'indépendance conditionnelle est donnée à l'avance. D'après la relation des modèles graphiques gaussiens (1.3), pour toutes variables $X_i$ et $X_j$ indépendantes conditionnellement au reste du processus, $k_{ij} = 0$ où $K = [k_{ij}]$ est la matrice de précision, inverse de la matrice de covariance $\Sigma$. C'est cette relation qui est utilisée pour générer des processus gaussiens dont la structure d'indépendance conditionnelle est connue. Une fois la matrice $K$ générée, le processus $\mathbf{X}$ est simulé : $\mathbf{X} \sim \mathcal{N}_p(\mathbf{0}, K^{-1})$ où $p$ est le nombre de nœuds du graphe $G$ représentant la structure d'indépendance conditionnelle choisie.

La principale difficulté rencontrée pour générer des processus multivariés gaussiens dont la structure d'indépendance conditionnelle est connue réside dans la simulation de la matrice $K$. En effet comme la matrice $K$ est censée être une matrice de précision, elle doit être symétrique définie positive et comme nous imposons la structure d'indépendance conditionnelle, elle doit respecter cette structure en ayant des valeurs nulles là où la structure n'a pas d'arêtes et des valeurs non nulles ailleurs. Nous considérons qu'une valeur $k_{ij}$ est significativement non nulle quand elle permet d'obtenir une valeur de corrélation partielle $\pi_{ij} = -k_{ij}/\sqrt{k_{ii}k_{jj}}$ significativement non nulle.

Deux approches sont utilisées dans la littérature pour générer une matrice $K$ qui est à la fois symétrique définie positive et associée à une structure d'indépendance conditionnelle donnée : soit elle est imposée, soit elle est simulée de façon à répondre à certains critères selon les méthodes de simulation.

### 2.1.1 Choix de la matrice de précision - Étude bibliographique

#### 2.1.1.1 Matrice de précision fixée

Il est important de noter que lorsqu'une étude sur des données simulées est présentée, elle est généralement faite sur quelques exemples n'illustrant qu'une infime partie des cas potentiellement observables. En effet, la (ou les) matrice(s) de précision $K$ utilisée(s) pour générer un processus gaussien multivarié sont fixées. Cette approche permet de connaître les performances de la méthode uniquement sur des cas particuliers.

Nous présentons quelques exemples des matrices utilisées dans la littérature. Nous notons la matrice de covariance $\Sigma = [\sigma_{ij}]$ et son inverse la matrice de précision $K = [k_{ij}]$. Certains exemples font intervenir des modèles auto-régressifs (AR). Dans notre contexte de simulation de matrices, un modèle AR(i) désigne une matrice ayant des valeurs non nulles uniquement sur la diagonale et sur les $i$ diagonales inférieures et supérieures. Par exemple, un modèle AR(1) est une matrice tridiagonale. En aucun cas le temps n'est utilisé dans ces simulations. Voici une liste non exhaustive d'exemples rencontrés dans la littérature :

- Carvalho et Scott [CS09] utilisent un modèle AR(10) pour des données de taille $p = 50$.
- Wang [Wan12] utilise 6 modèles pour $(p, n) = (30, 50)$ ou $(100, 200)$ :
  - un modèle AR(1) avec $\sigma_{ij} = 0.7^{|i-j|}$,

- un modèle AR(2) avec $k_{ii} = 1$, $k_{i,i-1} = k_{i-1,i} = 0.5$ et $k_{i,i-2} = k_{i-2,i} = 0.25$,
- un modèle par bloc avec $\sigma_{ii} = 1$, $\sigma_{ij} = 0.5$ pour $1 \leq i \neq j \leq p/2$ et $p/2+1 \leq i \neq j \leq 10$,
- un modèle en étoile autour du nœud 1 avec $k_i i = 1$, $k_{1,i} = k_{i,1} = 0.1$,
- un modèle circulaire avec $k_{ii} = 2$, $k_{i,i-1} = k_{i-1,i} = 1$ et $k_{1p} = k_{p1} = 0.9$,
- un modèle complet avec $k_{ii} = 2$ et $k_{ij} = 1$ pour $i \neq j$.

Ces exemples ne sont qu'un échantillon des exemples rencontrés mais ils illustrent le type de matrices de précision utilisées.

#### 2.1.1.2 Matrice de précision simulée

Dans le cas où $K$ n'est pas fixée, nous avons trouvé dans la littérature deux types d'approches qui permettent de simuler une matrice de précision $K$ associée à une structure d'indépendance conditionnelle représentée par le graphe $G$ :

1. la première génère une matrice $K$ parcimonieuse, simulée en imposant le nombre de valeurs nulles mais pas la position de ces valeurs. Le graphe $G$ est ensuite construit à partir de $K$. C'est le cas de la méthode de simulation utilisée par Giraud *et al.* [GHV12] (paquet R GGMselect) ou par Banerjee *et al.* [BGd08] et Fan *et al.* [FFW09]. L'inconvénient de ce type d'approches est que nous ne pouvons pas contrôler la structure d'indépendance conditionnelle de la matrice qui va être générée.
2. pour le second type d'approches, le graphe $G$ est donné et $K$ est simulée de façon à respecter $G$. Cette approche est utilisée par Donnet et Marin [DM12] et par Castelo dans [CR06] [CR09] (paquet R *qpgraph*). Elle a pour avantage de pouvoir imposer à la matrice $K$ la structure d'indépendance conditionnelle choisie. Par contre ces méthodes ne garantissent pas que les valeurs non nulles de la matrice de précision $K$ sont significativement non nulles.

Ces méthodes sont détaillées dans la suite.

##### 2.1.1.2.1 Méthodes où la structure d'indépendance conditionnelle est générée aléatoirement

La méthode la plus simple est présentée notamment dans [BGd08]. Dans un premier temps, ils génèrent aléatoirement une matrice $K_0$ diagonale dont les entrées sur la diagonale sont positives. Ensuite, ils ajoutent aléatoirement des éléments non nuls hors diagonale et de façon à avoir une matrice symétrique. Pour être sûrs que la matrice est définie positive, ils ajoutent un multiple de la matrice identité, ce multiple impose que la plus petite valeur propre de la matrice simulée est positive : $K = K_0 + \kappa I$ où $I$ est la matrice identité et $\kappa > |\min(0, \lambda_{min})|$, $\lambda_{min}$ étant la plus petite valeur propre de $K_0$. Cette approche permet de contrôler le taux de parcimonie mais ne contrôle ni la position des valeurs nulles et aucune information n'est donnée sur les intervalles dans lesquels sont choisies les valeurs non nulles. De plus, assurer le caractère défini positif de $K$ en ajoutant un multiple de la matrice identité peut faire tendre certaines valeurs de corrélation partielle non nulles vers 0 et ainsi changer le taux de parcimonie de la structure.

Une autre approche assez similaire est présentée dans [FFW09], inspirée de la méthode proposée dans [LG06]. Cette méthode, contrairement à la méthode précédente, contrôle le degré de chaque nœud du graphe associé à la matrice simulée. En effet, ils génèrent aléatoirement un certain nombre de points dans un carré unité et ils calculent les distances entre chaque paire de points. Chaque point est connecté aux $d$ points les plus proches. Ils font varier la valeur de $d$ pour faire varier le degré de parcimonie des graphes générés. Une fois la structure construite, chaque valeur non nulle est générée aléatoirement selon une loi uniforme sur $[-1, -0.5] \cup [0.5, 1]$ et $k_{ii} = 2 \sum_{j=1, j\neq i}^{j=p} |k_{ij}|$, le multiple 2 permet d'assurer le caractère défini positif de la matrice simulée. Cette méthode permet de générer des matrices $K$ dont la structure d'indépendance conditionnelle a le même degré $d$ sur chacun de ses nœuds. Nous ne pouvons pas contrôler

d'autres aspects de la structure. De plus, en assurant le caractère défini positif de $K$ en posant $k_{ii} = 2\sum_{j=1,j\neq i}^{j=p} |k_{ij}|$, certaines valeurs de corrélation partielle non nulles risquent de tendre vers 0 et ainsi faire décroître le degré de certains nœuds.

Une approche plus complexe est celle proposée par Giraud *et al.* [GHV12]. Cette méthode permet de contrôler le degré de parcimonie de la structure d'indépendance conditionnelle simulée aussi bien localement (sur un groupe de nœuds ⇔ sur une sous-matrice de $K$) que globalement (sur l'ensemble de la structure ⇔ sur la matrice $K$). Elle se compose des étapes suivantes :

1. Simulation globale de la matrice de précision $K$ :
   - tirage du nombre d'arêtes selon une loi binomiale de probabilité $\eta_{extra}$
   - sélection aléatoire des arêtes présentes (sélection uniforme sur les arêtes pas encore choisies)
   - tirages des valeurs non nulles de $K$ (correspondant aux arêtes) uniformément sur l'intervalle $[-1, 1]$. Seule la partie triangulaire inférieure $K_{inf}$ est conservée.
2. Simulations locales de la matrice de précision : création de trois sous-groupes de nœuds dont taille la taille est égale à la partie entière de $\frac{p}{3}$. Pour chaque sous-groupe :
   - tirage du nombre d'arêtes selon une loi binomiale de probabilité $\eta$
   - sélection aléatoire des arêtes présentes (sélection uniforme sur les arêtes pas encore choisies)
   - tirages des valeurs non nulles de $K$ (correspondant aux arêtes) uniformément sur l'intervalle $[-1, 1]$. Seule la partie triangulaire inférieure $K_{inf}$ est conservée.
3. les valeurs de la diagonale de $K_{inf}$ sont simulées selon une loi uniforme sur $]0, 1/10[$. Une matrice diagonale $D$ est aussi simulée : ses valeurs non nulles sont générées aléatoirement selon une loi uniforme sur $[\frac{1}{1000}, \frac{5}{1000}]$.
4. $K = K_{inf} \times K_{inf}^t \times 10 + D$, le produit de $K_{inf}$ par sa transposée impose que la matrice générée est semi-définie positive. De plus toutes les valeurs sur la diagonale sont non nulles, alors $K_{inf}$ est inversible et $K$ est définie positive.

Pour toutes ces méthodes, la structure de graphe $G$ est générée soit de façon totalement aléatoire, soit de façon à respecter certains critères comme le degré des nœuds pour la méthode de [FFW09]. L'inconvénient majeur de ces méthodes est qu'elles permettent difficilement de générer des matrices $K$ distinctes mais ayant la même structure d'indépendance conditionnelle. Ainsi, ces méthodes de simulations ne permettent pas de tester la reproductibilité d'une méthode d'estimation de modèles graphiques gaussiens pour une structure d'indépendance conditionnelle donnée. En effet, si pour une structure d'indépendance conditionnelle donnée, tous les processus générés sont générés à partir de la même matrice de précision alors le risque est de tester l'impact de la matrice de précision plutôt que celui de la structure.

#### 2.1.1.2.2 Méthodes où la structure d'indépendance conditionnelle est choisie

D'autres méthodes de simulation permettent de générer des matrices de précision respectant une structure d'indépendance conditionnelle $G$ donnée par l'utilisateur. Puisque ces méthodes permettent de générer plusieurs matrices de précisions distinctes à partir de la même structure d'indépendance conditionnelle, il est alors possible de générer des processus ayant la même structure d'indépendance conditionnelle mais étant générés à partir de matrices de précision distinctes. Ainsi, ces méthodes permettent de tester la reproductibilité des méthodes d'estimation de modèles graphiques gaussiens pour une structure de graphe donnée sans être impacté par le choix de la matrice de précision.

Travaillant sur les graphes décomposables, Donnet et Marin [DM12] peuvent simuler directement à partir d'une loi hyper-Inverse Wishart la matrice de covariance dont l'inverse respecte la

structure de graphe $G$ donnée par l'utilisateur, sous réserve que $G$ soit décomposable. Ils choisissent comme paramètres de la loi hyper-Inverse Wishart la matrice d'échelle $\tau I$ et $\delta = 1$ comme degré de liberté. $I$ est la matrice identité de taille $p \times p$. Ils font varier le paramètre $\tau$ pour leurs simulations.

Cette méthode a été développée pour tester les performances d'une méthode d'estimation des modèles graphiques gaussiens qui se concentre sur l'estimation du paramètre de la loi Wishart $\tau$ et du paramètre de la distribution de Bernoulli, a priori sur $G$. Cependant, elle peut être utilisée pour générer des processus pour tester d'autres méthodes d'estimation. Pour les méthodes de type bayésien, il faut veiller à ce que les paramètres utilisés pour générer les processus soient en accord avec les a priori choisis dans la méthode d'estimation, à moins que l'objectif soit d'étudier le comportement de la méthode quand les a priori ne correspondent pas aux processus, ce qui est classique lorsque les données étudiées sont des données réelles.

Pour les graphes non-décomposables, Castelo [CR09] propose dans le paquet R *qpgraph* une fonction permettant de générer une matrice de covariance dont l'inverse $K$ respecte la structure $G$ donnée par l'utilisateur. Cette fonction est composée des étapes suivantes :

1. Les matrices $D = \frac{1}{\sqrt{p}} I$ et $P = \rho I$ sont générées. $\rho$ est la moyenne de la corrélation marginale et vaut 0.5 par défaut et $I$ est la matrice identité de taille $p \times p$.
2. la matrice $W$ est simulée aléatoirement selon une loi Wishart avec pour matrice d'échelle $DPD$ et $p$ degrés de liberté.
3. l'algorithme de Hastie, Tibshirani et Friedman ([HTF09], chapitre 17, algorithme 17.1) est ensuite utilisé pour générer $\Sigma$ en prenant $W$ comme matrice empirique et $G = (V, E)$ comme graphe. Cet algorithme de Hastie, Tibshirani et Friedman permet de générer une matrice de covariance $\Sigma$ telle que $(\Sigma^{-1})_{a,b} = 0$ si $(a, b) \notin E$ et $\Sigma_{a,b} = W_{a,b}$ si $(a, b) \in E$. Nous reviendrons dans le chapitre 4 sur les raisons pour lesquelles cet algorithme a été crée.

#### 2.1.1.3 Avantages et inconvénients des méthodes existantes

Parmi les méthodes présentées ci-dessus, la méthode de Castelo est la plus adaptée à l'étude que nous souhaitons réaliser car elle permet de simuler des matrices de covariance dont l'inverse respecte la structure de graphe $G$ donnée en entrée et est applicable à tous les graphes. Cependant, aussi bien dans cette méthode que dans les autres, il n'existe aucun contrôle de la valeur minimale, en valeur absolue, des valeurs de corrélation partielle non nulles. En effet, pour un nombre de variables $p$ et d'observations $n$, la valeur à partir de laquelle une corrélation est significative, c'est-à-dire est statistiquement différente de zéro, varie (cf [HB11]). Cela est également valable pour les corrélations partielles. Connaissant $p$ et $n$, nous souhaitons simuler des matrices de corrélations partielles où les valeurs non-nulles sont significatives. Rappelons que la matrice de corrélations partielles $\Pi$ est directement liée à la matrice de précision $K$ : pour $i \neq j$, $\pi_{ij} = -k_{ij}/\sqrt{k_{ii}k_{jj}}$ et $\pi_{ii} = 1$.

### 2.1.2 Nouvelle méthode de simulation de processus tests

L'objectif de la méthode de simulation que nous proposons est de générer une matrice des corrélations partielles (reliée à la matrice de précision comme vu section 1.1.1.2 et rappelé ci-dessus) qui respecte une structure d'indépendance conditionnelle $G$ donnée et dont les valeurs non nulles sont significatives. Une valeur est considérée comme significativement non nulle quand l'hypothèse selon laquelle cette valeur est nulle a une probabilité très faible. Il existe un seuil $s$, qui dépend du nombre de variables $p$ et du nombre d'observations $n$ du processus associé à la matrice des corrélations partielles, pour lequel toute corrélation partielle supérieure en valeur

absolue à ce seuil est considérée comme significativement non nulle. Nous présentons d'abord l'algorithme de simulation en fixant le seuil $s$ et nous expliquons ensuite comment fixer ce seuil.

#### 2.1.2.1 Algorithme

Notre méthode est composée des étapes suivantes :

1. nous générons la matrice d'adjacence de $G$ notée $A$ : la matrice $A$ contient des 1 là où $G$ a des arêtes et des zéros ailleurs
2. nous simulons une matrice des corrélations partielles théorique $\Pi$ : $\forall i < j$
   - si $a_{ij} = 0$, $\pi_{ij} = 0$
   - si $a_{ij} = 1$, $\pi_{ij}$ est choisi aléatoirement selon une loi uniforme sur l'intervalle $[-1, -s] \cup [s, 1]$
3. la matrice est rendue symétrique : $\pi_{ij} = \pi_{ji}$ et $\pi_{ii} = 1$.
4. nous vérifions si la matrice est définie positive (si le minimum de ses valeurs propres est supérieur à 0)
   - si la matrice est définie positive, $\Pi$ est inversée pour donner $\Sigma$
   - sinon, il faut recommencer à l'étape 2.

La faiblesse de cette approche est qu'elle peut nécessiter de générer beaucoup de matrices avant d'en obtenir une définie positive. Cela est problématique pour générer des matrices dans le cas de données en grande dimension car pour certaines structures de graphe $G$ la proportion de matrices définies positives respectant $G$ et ayant des valeurs non nulles hors diagonale supérieures en valeur absolue à $s$ est très faible. Prenons par exemple un processus de $p = 12$ variables, nous souhaitons qu'il ait pour structure d'indépendance conditionnelle une structure en boucle, c'est-à-dire $(i, j) \in E \Leftrightarrow j = i + 1 \leq p$ ou $i = 1$ et $j = p$. Pour $s = 0.2$, parmi les matrices respectant cette structure grâce à la relation de modèles graphiques gaussiens, moins de 0.5% sont définies positives. Cette valeur a été obtenue en faisant des simulations numériques : pour avoir 1000 matrices respectant la structure de graphe imposée et étant définies positives en utilisant notre algorithme, il faut générer environ 212000 matrices respectant la structure imposée.

> Information Toolbox : notre algorithme de simulation de processus multivariés gaussiens dont la structure d'indépendance conditionnelle est connue est présent dans notre Toolbox. La fonction *simu_mat* génère une matrice des corrélations partielles respectant une structure donnée sous forme de matrice d'adjacence et dont les valeurs non nulles sont supérieures ou égales en valeur absolue à un seuil $s$ donné. Cette fonction donne également la matrice de covariance $\Sigma$ associée à la matrice des corrélations partielles générée. La fonction *simu_data* génère un processus multivarié gaussien ayant une moyenne nulle, de covariance $\Sigma$ et de $n$ observations.

#### 2.1.2.2 Choix du seuil $s$

Nous avons choisi de générer les éléments hors-diagonale non nuls sur l'intervalle $[-1, -s] \cup [s, 1]$ avec $s$ choisi pour garantir que les valeurs de corrélations partielles non nulles sont significativement non nulles. Pour choisir $s$, nous faisons un test statistique. Nous voulons que ce seuil $s$ soit supérieur à la valeur pour laquelle la corrélation partielle $\pi_{ij}^{emp}$ entre deux variables d'un processus gaussien à $p$ variables et $n$ observations est considérée significativement différente de zéro. Nous formulons le test ainsi :

$$\mathcal{H}_0 : \pi_{ij} = 0 \Leftrightarrow \pi_{ij}^{emp} < s_{min}$$
$$\mathcal{H}_1 : \pi_{ij} \neq 0 \Leftrightarrow |\pi_{ij}^{emp}| \geq s_{min}$$

Nous utilisons la transformée de Fisher :

$$z^{emp} = \frac{1}{2}\ln\left(\frac{1+\pi_{ij}^{emp}}{1-\pi_{ij}^{emp}}\right) \text{ et } \zeta = \frac{1}{2}\ln\left(\frac{1+s_{min}}{1-s_{min}}\right)$$

Nous avons vu (section 1.2.1) que

$$(z^{emp} - 0)\sqrt{n_p} \overset{a}{\sim} \mathcal{N}(0,1)$$

avec $n_p = n - 3 - (p-2)$.
Nous voulons trouver $\zeta$ tel que si $z < \zeta$, $\mathbb{P}(|(z^{emp}-0)\sqrt{n_p}| \leq |\zeta\sqrt{n_p}|)$ soit très proche de 1. Cela équivaut à vouloir un risque de faux positif $\alpha$ très faible. Nous fixons arbitrairement $\alpha = 0.001$

Nous souhaitons que $\mathbb{P}(|(z^{emp}-0)\sqrt{n_p}| \leq \zeta\sqrt{n_p}) = 1-\alpha$ ce qui équivaut à $2\Phi(\zeta\sqrt{n_p}) - 1 = 1-\alpha$. D'après les tables de la loi normale, cela revient à choisir $\zeta = 3.29/\sqrt{n_p}$.

Par transformée de Fisher inverse, nous obtenons :

$$s_{min} = \frac{\exp(2\frac{3.29}{\sqrt{n_p}}) - 1}{\exp(2\frac{3.29}{\sqrt{n_p}}) + 1}$$

Par exemple, pour un processus gaussien multivarié de 6 variables et 600 observations, le seuil $s$ doit être supérieur à 0.134 et pour $n = 60$ à 0.424.

Information Toolbox : dans notre Toolbox, la fonction *significant_ thresh* donne le seuil à partir duquel une corrélation partielle est statistiquement non nulle en valeur absolue sachant le nombre de variables $p$ et le nombre d'observations $n$.

## 2.2 Mesures de performances pour les méthodes d'extraction de structure d'indépendance conditionnelle

Sachant que nous savons générer des processus synthétiques dont nous connaissons la structure d'indépendance conditionnelle et que nous souhaitons évaluer les performances d'estimation de ces structures pour différentes méthodes, il faut pouvoir évaluer la qualité de la solution obtenue sachant la structure attendue. Pour cela nous introduisons différentes métriques.

### 2.2.1 Mesures utilisées

Il existe principalement trois catégories de mesures utilisées dans le cadre de l'évaluation des performances de méthode d'estimation de modèles graphiques gaussiens :

- les métriques permettant d'évaluer des paramètres estimés par la méthode
- les métriques permettant d'évaluer les performances de la méthode sur la matrice estimée dans le cas de méthodes estimant une matrice
- les métriques permettant d'évaluer les performances de la méthode sur graphe estimé comme représentant de la structure d'indépendance conditionnelle.

Concernant la première catégorie, nous pouvons prendre l'exemple de l'étude de Donnet et Marin [DM12]. Dans cette étude, l'estimation porte sur les paramètres $r$, régulant la probabilité d'une arête, et $\tau$ qui définit la matrice d'échelle $A$ utilisée pour générer la matrice de covariance du processus à partir d'une loi Wishart : $A = \tau I_p$. Puisque l'objectif n'est pas d'estimer au mieux la structure d'indépendance conditionnelle mais les paramètres $r$ et $\tau$, aucune métrique sur les graphes n'est utilisée.

Concernant la deuxième catégorie, les méthodes comme les méthodes de pénalisation ont pour objectif d'estimer la qualité de la matrice de précision estimée donc les performances de la méthode sont mesurées en terme de distances entre les matrices et non sur les graphes. Les métriques utilisées sont généralement des fonctions de coût (fonctions de coût entropique, quadratique, $\ell_1$, ...). Par exemple, Wang [Wan12] utilise le ratio entre la norme $\ell_1$ de la matrice de précision estimée par son approche bayésienne du Graphical lasso et la norme $\ell_1$ de la matrice de précision empirique.

Concernant les méthodes s'intéressant à la structure d'indépendance conditionnelle, il existe peu de mesures car peu d'auteurs utilisent des données synthétiques. Quand c'est le cas, les métriques utilisées sont la sensibilité, la spécificité et le coefficient de corrélation de Matthews que nous explicitons dans la suite. Ces métriques permettent de comparer la structure estimée par rapport à la structure attendue.

La sensibilité et la spécificité sont deux mesures statistiques utilisées depuis longtemps pour caractériser un résultat binaire (vrai/faux par exemple en épidémiologie ou en analyse statistique). En épidémiologie, la sensibilité représente la proportion de personnes malades identifiées comme malades et la spécificité la proportion de personnes saines identifiées comme saines. Fan *et al.* [FFW09] introduisent ces mesures pour l'estimation de graphes en posant que la sensibilité est la proportion d'arêtes attendues bien estimées et la spécificité la proportion d'arêtes non-attendues détectées comme absentes. Ils ajoutent à ces deux mesures le coefficient de corrélation de Matthews (CCM), très utilisé en Machine Learning pour mesurer la qualité d'un classifieur binaire. Cette mesure donne un résultat sur les performances générales de la méthode.

Nous notons $VP$ le nombre d'arêtes présentes correctement estimées, $FN$ le nombre d'arêtes présentes mal estimées, $VN$ le nombre d'arêtes absentes bien estimées et $FP$ le nombre d'arêtes absentes mal estimées. Ces notations sont organisées dans la matrice de confusion (table 2.1).

| | condition | | |
|---|---|---|---|
| population totale | arête présente | arête absente | |
| arête estimée présente | VP | FP | |
| arête estimée absente | FN | VN | |
| | sensibilité | spécificité | CCM |

TABLEAU *2.1 – Version simplifiée de la matrice de confusion dans le cadre de l'estimation des arêtes d'un graphe.*

Les mesures sont calculées de la manière suivante :

$$\text{sensibilité} = \frac{VP}{VP+FN} \qquad \text{spécificité} = \frac{VN}{VN+FP}$$

$$\text{CCM} = \frac{VP \times VN - FP \times FN}{\sqrt{(VP+FP)(VP+FN)(VN+FP)(VN+FN)}}$$

La sensibilité et la spécificité sont comprises entre 0 et 1, 0 pour une très mauvaise classification (aucun des éléments attendus n'est estimé dans la classe étudiée), 0.5 pour une classification similaire à une classification aléatoire et 1 pour une classification parfaite (tous les éléments attendus sont présents). Le CCM est compris entre -1 et 1, -1 correspondant à une inversion des deux classes, 0 à une classification aléatoire et 1 à une classification parfaite.

Ces mesures sont reprises par [Wan12] pour évaluer les performances de son approche bayésienne du Graphical lasso pour estimer la structure d'indépendance conditionnelle. Dans la suite nous retenons uniquement les mesures de sensibilité et de spécificité car la mesure CCM est

problématique dans le cas où tous les éléments sont estimés dans la même classe et est moins intuitive. Par exemple, dans le cadre de l'estimation d'une structure d'indépendance conditionnelle, si la structure obtenue est complète, c'est-à-dire que $FN = 0$ et $VN = 0$ alors nous devons diviser par la racine carrée de 0 ce qui n'est pas possible.

### 2.2.2 Mesure complémentaire

Nous cherchons à savoir si le graphe solution de la méthode considérée correspond à la structure d'indépendance conditionnelle que nous avons imposée au processus simulé observé. Comme nous nous concentrons sur des graphes non-dirigés et que seule la présence ou non d'une arête nous intéresse, nous pouvons travailler avec le support du graphe qui représente le graphe sous la forme d'un vecteur.

**Définition 2.2.1.** ***Distance de Hamming :*** *Soient $\mathcal{A}$ un alphabet et $\mathcal{E}_n$ l'ensemble des suites de longueur $n$ à valeur dans $\mathcal{A}$, la distance de Hamming $DH$ entre $u$ et $v$ deux éléments de $\mathcal{E}_n$ vaut :*

$$DH(u, v) = \sum_{i=1}^{n} u_i \oplus v_i$$

*où $\oplus$ désigne le "ou exclusif".*

La distance de Hamming entre deux graphes est la distance de Hamming entre les supports de ces graphes, les supports pouvant être vus comme des suites binaires de longueur $\frac{p(p-1)}{2}$. La distance de Hamming entre deux graphes correspond au nombre d'arêtes de différence entre deux graphes, c'est-à-dire le nombre d'arêtes présentes dans un graphe et pas dans l'autre. Elle est comprise entre 0 et $\frac{p(p-1)}{2}$, le nombre d'arêtes possibles dans un graphe à $p$ nœuds. Pour $p = 6$, la distance de Hamming est comprise entre 0 et 15.

La figure 2.1 illustre la nécessité de combiner la distance de Hamming avec la sensibilité et la spécificité pour juger de la pertinence d'un graphe. En effet, pour une même distance de Hamming, la sensibilité informe sur les arêtes potentiellement manquantes dans le graphe estimé et la spécificité sur les arêtes potentiellement en trop dans le graphe estimé.

Information Toolbox : la fonction *comp_ graphs* donne la sensibilité, la spécificité et la distance de Hamming entre deux graphes.

## 2.3 Proposition d'une procédure d'évaluation pour les méthodes d'estimation de structure d'indépendance conditionnelle

Pour évaluer une méthode, la première étape consiste à choisir aléatoirement plusieurs structures avec différentes caractéristiques : des graphes décomposables, des graphes non-décomposables, des graphes avec peu d'arêtes, des graphes avec beaucoup d'arêtes, des graphes avec des degrés similaires entre les nœuds et des graphes avec des degrés très hétérogènes, ... Ensuite, pour chaque structure il faut générer un ensemble test de matrices de précision $K$ (chacune associée à une matrice $\Pi$) respectant cette structure. Enfin, pour chaque matrice, on simule des processus gaussiens selon la loi $\mathcal{N}_p(\mathbf{0}, \Sigma)$ avec $\Sigma = \text{inv}(K)$ et de $n$ observations. Le nombre d'observations est à fixer en fonction des applications sur lesquelles la méthode veut être utilisée. Si l'objectif est d'appliquer la méthode sur des données réelles sous forme de processus de $p$ variables et $n$ observations, il faut tester les performances de la méthode sur des processus simulés ayant les mêmes caractéristiques.

La méthode à évaluer est appliquée aux différents processus. La qualité du résultat de la méthode est quantifiée en termes de distance de Hamming, de sensibilité et de spécificité pour

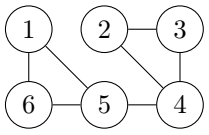
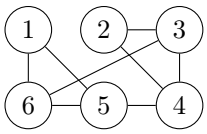
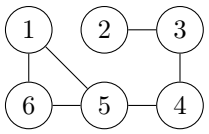
| | Graphe attendu :  | |
|---|---|---|
| $\widehat{G}$ |  |  |
| distance de Hamming | 1 | 1 |
| sensibilité | 1 | 0.86 |
| spécificité | 0.88 | 1 |

FIGURE *2.1 – Exemples montrant le comportement de la distance de Hamming et des mesures de sensibilité et spécificité. Les deux graphes estimés ont la même distance de Hamming avec le graphe attendu mais celui de gauche a une arête en plus alors que celui de droite a une arête en moins. Ces informations complémentaires sont données par les mesures de sensibilité et de spécificité : pour le graphe de gauche, la spécificité inférieure à 1 signifie qu'il y a des arêtes en trop par rapport au graphe attendu et pour le graphe de droite, la sensibilité inférieure à 1 signifie qu'il manque des arêtes par rapport aux arêtes attendues.*

les différentes structures. Si la distance de Hamming est proche de zéro et la sensibilité et la spécificité sont proches de 1, la méthode est plutôt efficace. La figure 2.2 illustre l'ensemble de la procédure d'évaluation.

Cette procédure permet de faire une évaluation quantitative des performances d'une méthode d'estimation de modèles graphiques gaussiens. Elle permet également de mener une étude comparative entre différentes méthodes d'estimation.

## 2.4 Conclusion

Dans ce chapitre, après avoir étudié différentes méthodes proposées dans la littérature pour simuler des processus dont la structure d'indépendance conditionnelle est connue, nous proposons une nouvelle approche pour simuler de tels processus. Cette méthode présente l'avantage de contrôler les valeurs minimales en valeur absolue des corrélations partielles non nulles afin que toutes ces corrélations soient statistiquement significatives. Bien que cette méthode ne soit pas optimisée et puisse être trop contraignante pour des données en "grande dimension", elle se révèle efficace pour traiter des problèmes de petite dimension que nous étudions dans la suite.

De plus, nous avons étudié les différentes mesures proposées dans la littérature pour évaluer la différence entre une structure d'indépendance conditionnelle estimée et celle fixée lors de la simulation du processus. Ces mesures comparent les graphes représentant les structures en terme de différence d'arêtes. Nous avons choisi de retenir la sensibilité et la spécificité auxquelles nous ajoutons la distance de Hamming. Ces mesures vont nous permettre de mesurer les performances des méthodes d'estimation de structure d'indépendance conditionnelle selon plusieurs critères : la distance entre la structure attendue et la solution obtenue mais aussi la qualité de l'estimation des arêtes attendues et des arêtes non attendues.

La méthode de simulation de processus gaussiens multivariés dont la structure d'indépen-

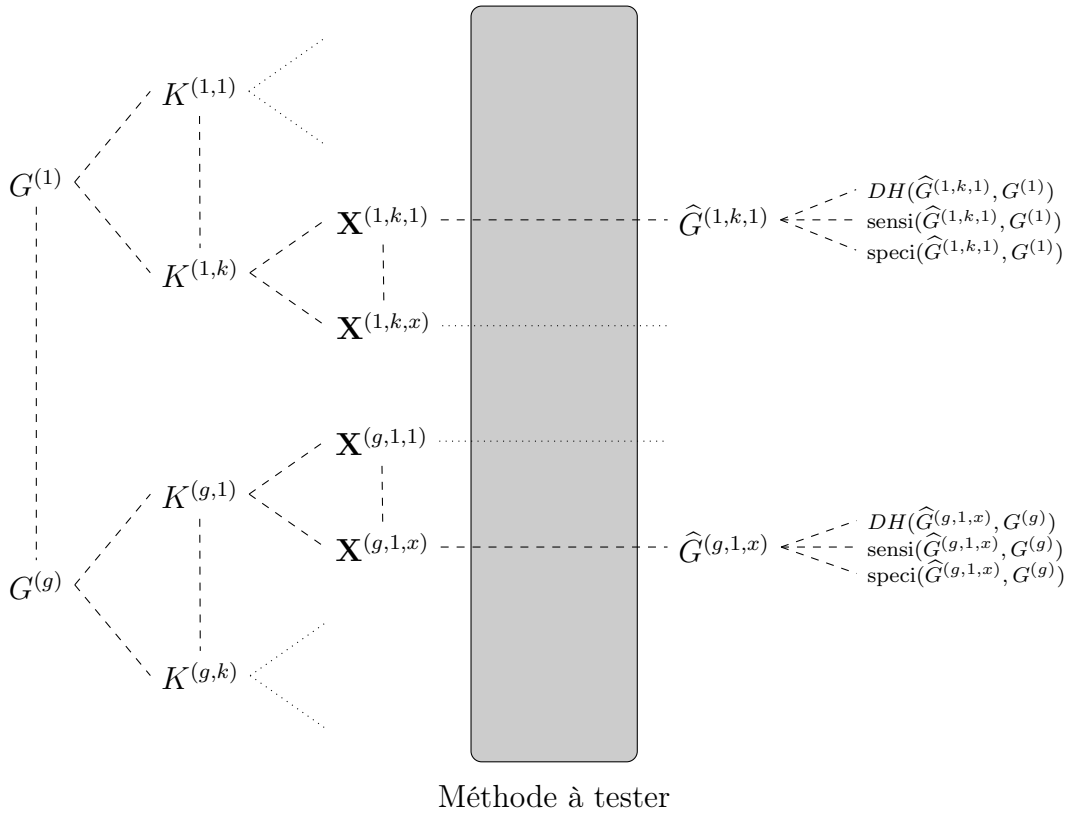



FIGURE *2.2 – Illustration de la procédure pour évaluer une méthode d'estimation de structure d'indépendance conditionnelle. K pourrait être remplacée par* $\Pi$ *de manière équivalente (c'est ce qui a été fait dans le cadre de cette thèse).*

dance conditionnelle est connue et les métriques de comparaison de graphes seront utilisées dans le chapitre 3 afin d'évaluer les performances d'estimation de la structure d'indépendance conditionnelle de la méthode que nous avons développée. Elles seront également utilisées dans le chapitre 4 où nous comparons les performances de notre méthode aux performances d'autres méthodes existantes.

# Chapitre 3

# Méthode ABiGlasso

**Sommaire**



Nous avons vu dans le chapitre 1 qu'il existe différentes méthodes pour estimer les structures d'indépendance conditionnelle dans le cadre des modèles graphiques gaussiens. Ces méthodes présentent toutes des avantages et des inconvénients variés comme, par exemple, le choix de paramètres difficiles mais la rapidité de calcul ou le temps de calcul très long voire rédhibitoire mais la possibilité de comparer la pertinence de plusieurs structures pour un processus donné. Dans

ce chapitre nous présentons la méthode ABiGlasso (Asymptotic Bayesian method initialized by Graphical lasso) qui se base sur une approche développée dans le cadre des méthodes bayésiennes pour pouvoir profiter de la comparaison entre structures et qui contient une étape d'initialisation qui utilise le Graphical lasso pour accélérer le temps de calcul. Le terme *asymptotic* dans le nom de la méthode fait référence au fait que nous utilisons une probabilité a posteriori qui est définie asymptotiquement.

Le chapitre est organisé de la façon suivante : une première partie présente la méthode et ses variantes, leurs motivations et leurs fonctionnements. Une deuxième partie porte sur l'étude théorique du temps de calcul de la méthode et de ses variantes. Les performances des différentes variantes sont ensuite étudiées empiriquement sur des processus simulés. Nous étudions, dans un premier temps, les performances des méthodes en fonction du nombre d'observations des processus pour évaluer sur quels types de processus nous pouvons appliquer nos méthodes et comment interpréter les résultats obtenus sachant le processus utilisé. Ensuite, nous étudions l'impact des paramètres d'entrée des méthodes sur la qualité de la solution obtenue. Nous étudions également l'influence de ces mêmes paramètres sur le temps de calcul avant de discuter des compromis entre temps de calcul et qualité de la solution obtenue. Au regard de ces différents aspects, nous terminons par une discussion sur la manière d'utiliser la méthode ABiGlasso.

## 3.1 La méthode

### 3.1.1 Motivation et principe

Parmi les méthodes présentées dans le chapitre 1, plusieurs méthodes ont retenu notre attention par leurs avantages. Les méthodes bayésiennes nous ont plus particulièrement intéressés car elles fournissent une probabilité a posteriori qui permet de comparer, pour un processus donné $\mathbf{X}$, la pertinence de différents graphes à représenter la structure d'indépendance conditionnelle de $\mathbf{X}$. Ce qui est intéressant dans cette approche, c'est que en plus d'avoir la structure d'indépendance conditionnelle la plus probable, nous pouvons savoir si cette structure est vraiment beaucoup plus probable que les autres structures ou si il existe un groupe de structures plus ou moins équiprobables qui représentent notre processus. Ce dernier cas revient à dire que la présence ou non d'un lien entre deux variables du processus peut ne pas être une information suffisante pour la description globale du processus. De plus, ne pas avoir juste une information de présence ou d'absence d'une arête mais une probabilité offre une plus grande richesse d'interprétation. Pour bénéficier de cet atout et ne souhaitant pas se limiter à un type de graphes, nous nous sommes concentrés sur les méthodes bayésiennes appliquées à l'ensemble des graphes sans restriction de structures. Nous nous sommes plus particulièrement focalisés sur le travail de Marrelec [MB06a] qui exprime la probabilité a posteriori d'un graphe $G$ sachant un processus $\mathbf{X}$, de $p$ variables et $n$ observations, de la façon suivante :

$$\mathbb{P}(G|\mathbf{X}) = C(\mathbf{X}) \times \frac{\varphi_{\boldsymbol{\pi}_{\bar{E}}, cW_{\bar{E}\bar{E}}}(\mathbf{0})}{V(G)} \tag{3.1}$$

Pour rappel, la constante $C(\mathbf{X})$ ne dépend pas du graphe considéré. $\varphi_{\mu,\Sigma}(\mathbf{y})$ est la loi gaussienne multivariée, de moyenne $\mu$, de covariance $\Sigma$ pour le vecteur $\mathbf{y}$. La constante $c = 1/(n+p+1)$, $W$ est la matrice d'Isserlis [RW98] de la matrice de précision empirique, $\boldsymbol{\pi}$ est le vecteur des corrélations partielles selon le support de $G$, les indices $\bar{E}$ et $\bar{E}\bar{E}$ indiquent que seules les valeurs associées aux arêtes absentes de $G$ sont considérées : $\bar{E}$ est le complémentaire de $E$ sur l'ensemble des arêtes possibles. $V(G)$ est le volume des matrices de corrélations partielles définies positives ayant des valeurs nulles là où le graphe $G$ n'a pas d'arêtes.

Le temps de calcul de $\mathbb{P}(G|\mathbf{X})$ dépend principalement du temps nécessaire à l'estimation de $V(G)$. Pour calculer $\mathbb{P}(G|\mathbf{X})$ pour l'ensemble des graphes, le temps de calcul devient très

important voire rédhibitoire. Par exemple, pour un processus $\mathbf{X}$ à $p = 6$ variables, il existe 32768 graphes possibles pouvant représenter sa structure d'indépendance conditionnelle, il faut donc calculer 32768 fois $V(G)$. Or $C(\mathbf{X})$ n'est obtenue que si les probabilités a posteriori de tous les graphes ont été calculées, c'est-à-dire qu'il faut calculer $V(G)$ pour tous les graphes. Pour s'affranchir de cette constante tout en gardant la possibilité de comparer la pertinence de différents graphes pour représenter la structure d'indépendance conditionnelle d'un processus, nous introduisons le score $z$ :

$$z(G|\mathbf{X}) = \frac{\varphi_{\boldsymbol{\pi}_{\bar{E}}, cW_{\bar{E}\bar{E}}}(\mathbf{0})}{V(G)}. \tag{3.2}$$

Le calcul de ce score $z$ est aussi dépendant de $V(G)$, mais nous n'avons plus besoin de calculer la constante $C(\mathbf{X})$ qui rendait l'utilisation de $\mathbb{P}(G|\mathbf{X})$ rédhibitoire.

Pour réduire les temps de calcul, nous souhaitons comparer un nombre réduit de graphes. Cette idée a été également utilisée par Giraud *et al.* [GHV12] qui travaillent sur des familles de graphes avec certaines propriétés. En ce qui nous concerne, nous souhaitons une approche rapide pour générer le sous-ensemble de graphes à explorer. Parmi les méthodes de la littérature, la méthode Graphical lasso [FHT08] est une méthode rapide qui estime une matrice de précision $\widehat{K}_\lambda$ parcimonieuse (c'est-à-dire avec certaines valeurs nulles) à partir de laquelle on peut construire un graphe $\widehat{G}_\lambda$ (les arêtes de $\widehat{G}_\lambda$ étant associées aux valeurs non nulles de $\widehat{K}_\lambda$). Pour rappel, le Graphical lasso est un algorithme qui estime la matrice de précision parcimonieuse vérifiant :

$$\widehat{K}_\lambda = \underset{K}{\operatorname{argmax}} \log(\det(K)) - \operatorname{tr}(K\Sigma^{emp}) - \lambda||K||_1 \tag{3.3}$$

où $K$ est la matrice de précision empirique de $\mathbf{X}$ et $\widehat{K}_\lambda$ une version parcimonieuse de $K$ dont le degré de parcimonie dépend de $\lambda$. Le graphe $\widehat{G}_\lambda$ est construit en mettant une arête entre $i$ et $j$ uniquement si $(\widehat{k}_\lambda)_{ij} \neq 0$. Cependant, cette méthode a deux principaux désavantages : le choix du paramètre de pénalisation $\lambda$ et la possibilité de non convergence du graphe $\widehat{G}_\lambda$ vers la structure d'indépendance conditionnelle attendue. Pour s'affranchir du choix du paramètre de pénalisation $\lambda$, nous choisissons de travailler sur l'ensemble des graphes générés à partir de l'algorithme Graphical lasso pour le vecteur $\Lambda = \{0, step_\Lambda, 2step_\Lambda, \ldots, \lambda_\emptyset\}$. Le nombre d'arêtes de $\widehat{G}_\lambda$ décroît quand $\lambda$ augmente, $\lambda_\emptyset$ est la valeur à partir de laquelle le graphe obtenu est vide. Concernant le problème de non convergence potentielle de l'algorithme de résolution de (3.3), nous choisissons d'étendre notre ensemble de graphes aux solutions du Graphical lasso et à leurs voisins.

**Définition 3.1.1.** ***Voisinage d'un graphe*** $G$ ***:*** *ensemble des graphes différant de $G$ d'au plus $e$ arêtes. $e$ est le paramètre de voisinage qui définit le nombre maximum d'arêtes de différence entre le graphe $G$ et un de ses voisins. Le voisinage est noté $\mathcal{G}_e(G)$.*

Notons $\gamma(e, p)$ le nombre d'éléments dans le voisinage de paramètre $e$ d'un graphe à $p$ nœuds, le sous-ensemble de graphes que nous avons choisi d'étudier a au plus $|\Lambda| \times \gamma(e, p)$ éléments. Il est important de noter que les voisinages de deux graphes distincts peuvent avoir plusieurs graphes en commun. Il faut également avoir à l'esprit que deux valeurs distinctes de $\Lambda$ peuvent donner le même graphe. En cas de redondance d'un graphe, nous ne le considérons qu'une seule fois : notre sous-ensemble de graphes ne contient que des éléments uniques.

La figure 3.1 donne deux exemples de voisinages pour des graphes à $p = 4$ nœuds pour $e = 1$. Pour deux valeurs distinctes du paramètre de pénalisation $\lambda$, $\lambda_1$ et $\lambda_2$, l'utilisation du Graphical lasso avec ces paramètres sur un même processus donne deux graphes distincts $\widehat{G}_{\lambda_1}$ et $\widehat{G}_{\lambda_2}$. Les voisinages de paramètre $e = 1$ de ces deux graphes sont constitués de 7 graphes : $\widehat{G}_\lambda$ et 6 graphes qui diffèrent tous d'une arête avec $\widehat{G}_\lambda$. Nous avons choisi cet exemple car il permet de représenter

facilement les deux voisinages et il met en évidence que l'intersection entre deux voisinages n'est pas toujours vide. Pour cet exemple, $\mathcal{G}_1(\widehat{G}_{\lambda_1}) \cap \mathcal{G}_1(\widehat{G}_{\lambda_2})$ contient deux graphes : celui pour lequel $E = \{(1,2),(2,3)\}$ et celui pour lequel $E = \{(1,2),(3,4)\}$ (en rouge dans la figure).

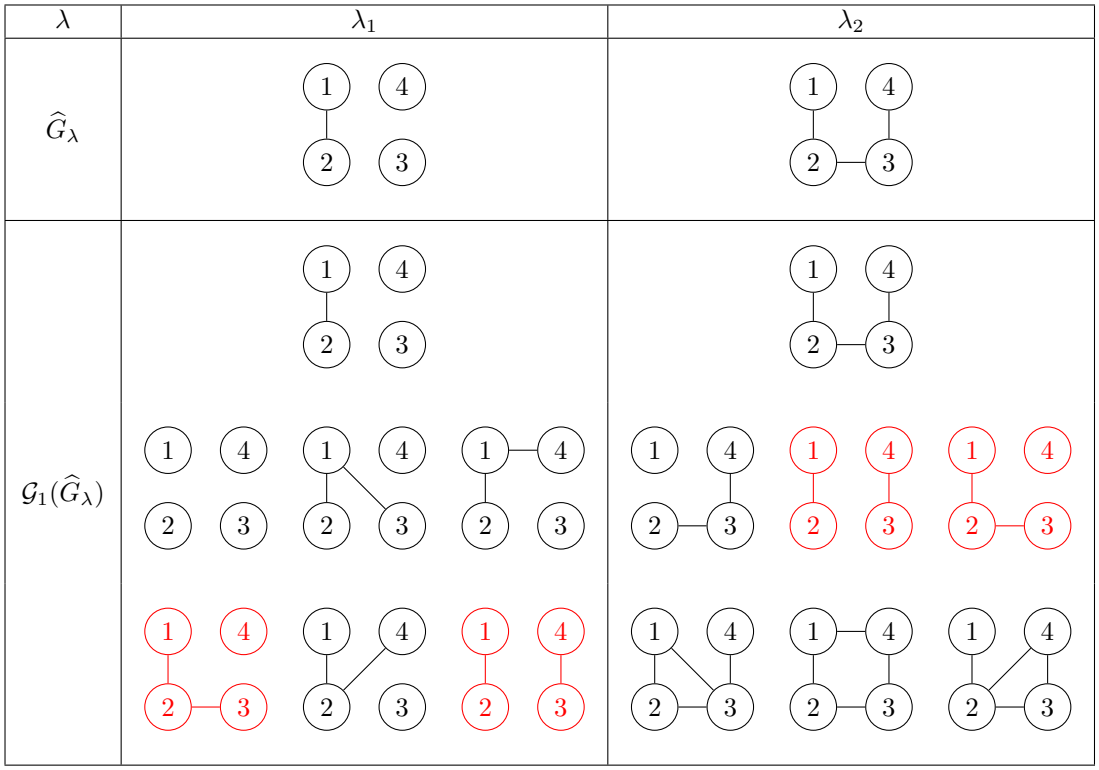



FIGURE *3.1 – Exemples de voisinages sur des graphes avec $p = 4$ nœuds et pour un paramètre de voisinage de $e = 1$. Pour deux $\lambda$ distincts, l'utilisation du Graphical lasso donne deux graphes $\widehat{G}_\lambda$ distincts. Les voisinages de ces deux graphes se composent de 7 graphes. De plus l'intersection de ces deux voisinages n'est pas vide, les graphes en rouge étant communs aux deux voisinages.*

Une fois le sous-ensemble de graphes obtenu, nous proposons d'en comparer les différents éléments en utilisant le score $z$ introduit précédemment (équation (3.2)).

Le score $z$ étant construit à partir d'une méthode bayésienne asymptotique et le Graphical lasso servant à l'initialisation du sous-ensemble de graphes, nous avons choisi d'appeler notre méthode ABiGlasso (Asymptotic Bayesian method initialized by Graphical lasso). Nous avons dérivé différentes versions de cette méthode qui sont présentées en détails dans la section suivante.

### 3.1.2 ABiGlasso et variantes

Dans cette partie, nous présentons en détail la méthode ABiGlasso et ses variantes. Les différentes variantes ont été introduites pour améliorer le temps de calcul et/ou les performances d'estimation de la structure d'indépendance conditionnelle.

#### 3.1.2.1 Méthode de base

Comme présentée dans la section précédente, la méthode ABiGlasso se compose de deux étapes :

1. Étape d'initialisation : Initialisation du sous-ensemble de graphes en utilisant l'algorithme Graphical lasso
2. Étape de comparaison : Comparaison des graphes en utilisant le score $z$ (cf équation (3.2)).

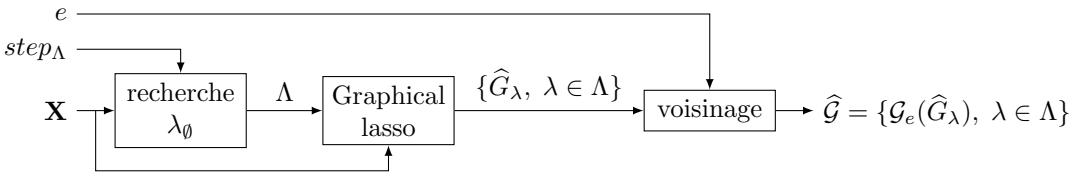
L'étape d'initialisation se compose elle-même de plusieurs étapes :

1. Elle commence par une recherche de la borne supérieure $\lambda_\emptyset$ de l'ensemble des paramètres de pénalisation $\Lambda$ à parcourir. Le nombre d'arêtes dans le graphe obtenu à partir du Graphical lasso varie en fonction de $\lambda$ : pour $\lambda = 0$, le graphe est complet puis le nombre d'arêtes décroît jusqu'à être nul, $\lambda_\emptyset$ est la valeur à partir de laquelle le graphe obtenu est vide. $\lambda_\emptyset$ est obtenu par une approche dichotomique : si $\widehat{G}_\lambda$ a un nombre d'arêtes non nul alors $\lambda_\emptyset > \lambda$ et si $\widehat{G}_\lambda$ n'a pas d'arête, $\lambda_\emptyset < \lambda$. On considère la convergence atteinte quand la différence entre les bornes supérieure et inférieure de $\lambda_\emptyset$ est plus petite que $step_\Lambda$. La sortie de cette étape est l'ensemble $\Lambda = \{0, step_\Lambda, 2step_\Lambda, \ldots, \lambda_\emptyset\}$.
2. Une fois $\Lambda$ obtenu, l'algorithme Graphical lasso est utilisé pour générer un graphe $\widehat{G}_\lambda$ par paramètre de pénalisation $\lambda \in \Lambda$. La version du Graphical lasso que nous utilisons est celle développée par Boyd [BPC+10]. La sortie de cette étape est l'ensemble $\{\widehat{G}_\lambda,\ \lambda \in \Lambda\}$. Notons que même si plusieurs valeurs $\lambda$ peuvent générer le même graphe, l'ensemble de sortie n'est composé que d'éléments distincts.
3. Pour chaque $G \in \{\widehat{G}_\lambda,\ \lambda \in \Lambda\}$, le voisinage $\mathcal{G}_e(G)$ est généré et les différents voisinages sont regroupés pour former le sous-ensemble de graphes à comparer dans l'étape de comparaison : $\widehat{\mathcal{G}} = \{\mathcal{G}_e(\widehat{G}_\lambda),\ \lambda \in \Lambda\}$. Notons ici aussi que même si deux voisinages peuvent avoir une intersection non vide, les éléments de $\widehat{\mathcal{G}}$ sont distincts.

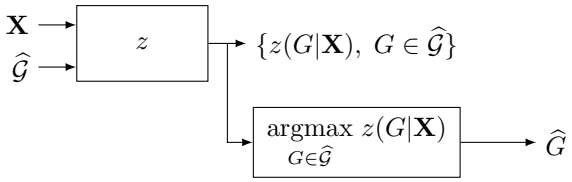
L'étape de comparaison consiste à appliquer le score $z$ (cf équation (3.2)) sur l'ensemble des graphes du sous-ensemble $\widehat{\mathcal{G}}$ obtenu à l'étape précédente. Une fois tous les scores calculés, il est possible d'obtenir une estimée de la structure d'indépendance conditionnelle de $\mathbf{X}$ en choisissant le graphe le plus probable sachant $\mathbf{X}$, c'est-à-dire ayant le score le plus élevé :

$$\widehat{G} = \underset{G \in \widehat{\mathcal{G}}}{\operatorname{argmax}}\ z(G|\mathbf{X}). \tag{3.4}$$

La figure 3.2 récapitule l'organisation de la méthode ABiGlasso de base : la figure 3.2a pour l'étape d'initialisation et la figure 3.2b pour l'étape de comparaison.




(a) Etape d'initialisation




(b) Etape de comparaison

FIGURE *3.2 – Schéma des deux étapes de la méthode ABiGlasso de base.*

La figure 3.3 illustre les différentes étapes de la méthode ABiGlasso de base sur un processus synthétique. Ce processus synthétique a été généré avec pour structure d'indépendance conditionnelle $G_{struct}$. Sur ce processus, la méthode de base avec pour paramètres $e = 3$ et $step_\Lambda = 0.1$ donne le résultat attendu. Cet exemple permet d'illustrer également plusieurs phénomènes mentionnés précédemment :

– plusieurs $\lambda \in \Lambda$ conduisent au même graphe : $|\Lambda| = 144$ et $|\{\widehat{G}_\lambda,\ \lambda \in \Lambda\}| = 12$.
– un graphe peut être présent dans les voisinages de différents graphes : si tous les voisinages des solutions du Graphical lasso sur $\Lambda$ n'avaient pas d'intersection, $\widehat{\mathcal{G}}$ aurait contenu 6912 graphes (contre 3973 quand les doublons sont enlevés).
– le Graphical lasso peut ne pas proposer le graphe solution : $\widehat{G} \notin \{\widehat{G}_\lambda,\ \lambda \in \Lambda\}$. Dans ce cas, $\widehat{G}$ est un voisin d'une solution du Graphical lasso. Ce phénomène peut également être dû au choix de $\Lambda$ : par exemple, pour une valeur plus faible de $step_\Lambda$, le Graphical lasso aurait pu proposer $\widehat{G}$.

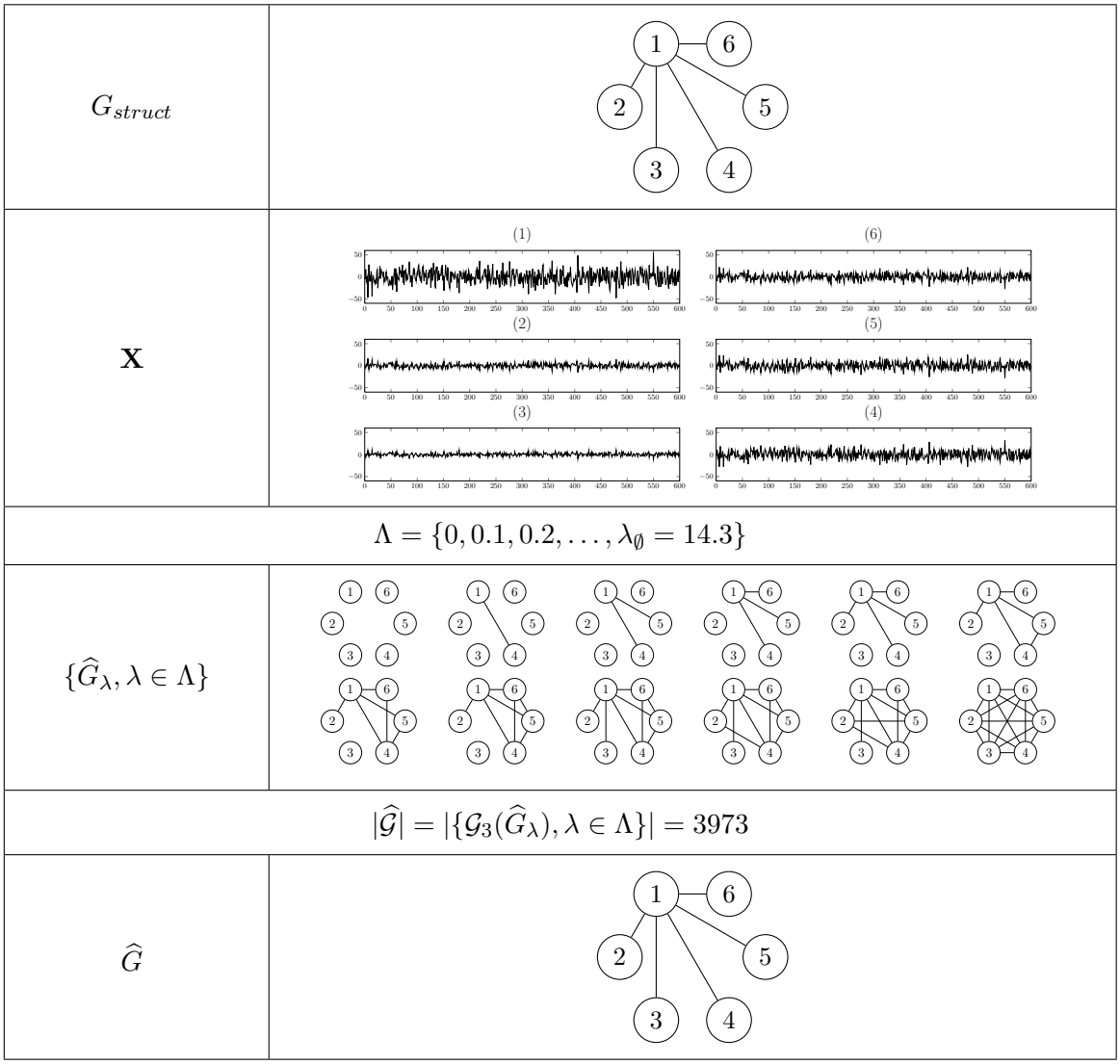



FIGURE *3.3 – Illustration des sorties des différentes étapes de la méthode ABiGlasso de base sur un processus synthétique. Ce processus synthétique a été généré avec pour structure d'indépendance conditionnelle $G_{struct}$. Sur ce processus, la méthode de base avec pour paramètres $e = 3$ et $step_\Lambda = 0.1$ donne le résultat attendu.*

L'étape d'initialisation de la méthode ABiGlasso génère en général un sous-ensemble de graphes avec un grand nombre d'éléments. Plus le nombre d'éléments est important, plus le temps de calcul est élevé. Nous proposons donc deux variantes de notre méthode pour réduire le nombre d'éléments de $\widehat{\mathcal{G}}$.

Information Toolbox : cette méthode est implémentée dans la Toolbox. C'est la fonction *Method_ ABiGlasso*.

#### 3.1.2.2 Variante 1 : Réduction de $\Lambda$ par approche empirique - ABiGlasso avec intervalle réduit

Suite à l'application de la méthode ABiGlasso de base avec $e = 3$ sur des processus à $p = 6$ variables, nous avons observé deux phénomènes :

- les graphes les plus probables sont en général des graphes qui diffèrent de moins de $e = 3$ arêtes avec un graphe $\widehat{G}_\lambda$ pour $\lambda \in \Lambda$ :
- pour des $\lambda$ très faibles, le voisin le plus probable de $\widehat{G}_\lambda$ a $e = 3$ arêtes de moins que $\widehat{G}_\lambda$ et pour des $\lambda$ proches de $\lambda_\emptyset$, le voisin le plus probable de $\widehat{G}_\lambda$ a $e = 3$ arêtes de plus. Ceci concorde avec le fait que le nombre d'arêtes de $\widehat{G}_\lambda$ diminue quand $\lambda$ augmente.

Ces observations sont à l'origine de l'idée de réduire $\Lambda$ à $\{\lambda_\infty^{min}, step_\Lambda, 2step_\Lambda, \dots, \lambda_\infty^{max}\}$ où $\lambda < \lambda_\infty^{min}$ implique que le graphe le plus probable dans le voisinage de $\widehat{G}_\lambda$ avec $e = 3$ a 3 arêtes de moins que $\widehat{G}_\lambda$ et où $\lambda > \lambda_\infty^{max}$ implique que le graphe le plus probable dans le voisinage de $\widehat{G}_\lambda$ avec $e = 3$ a 3 arêtes de plus que $\widehat{G}_\lambda$.

La figure 3.4 montre les observations que nous avons faites pour un processus. La courbe noire représente le score le plus élevé sur le voisinage de $\widehat{G}_\lambda$ en fonction de $\lambda$ : $\widehat{G}$ appartient aux voisinages de $\widehat{G}_\lambda$ avec $\lambda \in \{1.2, 1.4, 1.8, 2.2\}$. La courbe rouge représente $d(\lambda)$ :

$$d(\lambda) = |E(\widehat{G}_\lambda)| - |E\big(\operatorname*{argmax}_{G \in \mathcal{G}_e(\widehat{G}_\lambda)} z(G|\mathbf{X})\big)|. \tag{3.5}$$

$d(\lambda)$ donne la différence entre le nombre d'arêtes de $\widehat{G}_\lambda$ et de $\operatorname*{argmax}_{G \in \mathcal{G}_e(\widehat{G}_\lambda)} z(G|\mathbf{X})$. Si $d(\lambda)$ positif, c'est le graphe $\widehat{G}_\lambda$ qui a le plus d'arêtes, sinon c'est l'autre graphe.

En regardant simultanément les courbes noire et rouge, on constate que $\widehat{G}$ a une arête de plus que $\widehat{G}_\lambda$ pour $\lambda = 1.8$. Cela correspond à la première observation expliquée précédemment : $\widehat{G}$ diffère de moins de $e = 3$ arêtes avec $\widehat{G}_{1.8}$.

Le comportement de la courbe rouge correspond à la deuxième observation : pour $\lambda \leq 1.4$, $\operatorname*{argmax}_{G \in \mathcal{G}_3(\widehat{G}_\lambda)} z(G|\mathbf{X})$ a $e = 3$ arêtes de moins que $\widehat{G}_\lambda$ et pour $\lambda \geq 2$, il a $e = 3$ arêtes de plus que $\widehat{G}\lambda$.

Les pointillés en bleu mettent en évidence l'intervalle que nous souhaitons obtenir.

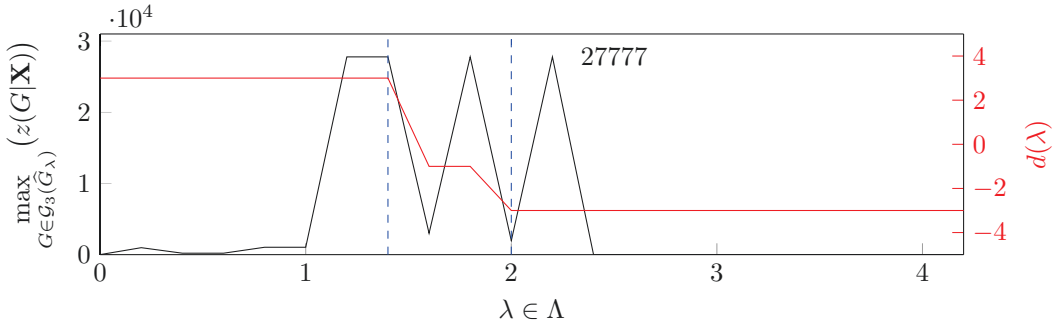



FIGURE *3.4 – En noir, le score le plus élevé sur le voisinage de $\widehat{G}_\lambda$ en fonction de $\lambda$, en rouge la différence d'arêtes entre $\widehat{G}_\lambda$ et son voisin le plus probable et les pointillés bleus délimitent le nouvel intervalle que nous souhaitons obtenir. Pour cette figure $e = 3$ et $step_\Lambda = 0.2$.*

Pour obtenir ce nouvel intervalle, nous procédons par dichotomie sur la distance $d(\lambda)$ (équation (3.5)). Nous cherchons les deux bornes de l'intervalle en parallèle, $\lambda^{min}$ sur $\left[l_g^{min}, l_d^{min}\right]$ et $\lambda^{max}$ sur $\left[l_g^{max}, l_d^{max}\right]$. L'algorithme converge en un temps fini vers $\left[\lambda_\infty^{min}, \lambda_\infty^{min}\right]$. Pour chaque itération :

- Pour la borne inférieure, si le voisin de $\widehat{G}_{\lambda^{min}(i)}$ ayant le score le plus élevé a $e$ arêtes de moins que $\widehat{G}_{\lambda^{min}(i)}$ alors $\lambda_\infty^{min} > \lambda^{min}(i)$ sinon $\lambda_\infty^{min} < \lambda^{min}(i)$.
- Pour la borne supérieure, si le voisin de $\widehat{G}_{\lambda^{max}(i)}$ ayant le score le plus élevé a $e$ arêtes de plus que $\widehat{G}_{\lambda^{max}(i)}$ alors $\lambda_\infty^{max} < \lambda^{max}(i)$ sinon $\lambda_\infty^{max} > \lambda^{max}(i)$.

Les deux bornes $\lambda_\infty^{min}$ et $\lambda_\infty^{max}$ sont estimées indépendamment l'une de l'autre. Nous considérons que la convergence est atteinte pour une borne quand $l_d - l_g < 0.02$.

La figure 3.5 illustre la recherche par dichotomie de la borne inférieure sur un exemple synthétique où la solution attendue est $\lambda_\infty^{min} = 7$, pour $e = 3$. La vignette (A) illustre la convergence de $\lambda^{min}$ en fonction des itérations de la recherche par dichotomie et la vignette (B) représente les variations de $d(\lambda^{min})$ en fonction de ces mêmes itérations. A la première itération, $d\left(\lambda^{min}(1)\right) = 3$ ce qui signifie que $\lambda^{min}(1) < \lambda_\infty^{min}$. Cela implique la mise à jour suivante : $l_g^{min}(2) = \lambda^{min}(1)$. A la deuxième itération, $d\left(\lambda^{min}(2)\right) = -2$ et donc $\lambda^{min}(2) > \lambda_\infty^{min}$ d'où la mise à jour suivante : $l_d^{min}(3) = \lambda^{min}(2)$. Le principe est le même jusqu'à ce que la convergence soit atteinte.

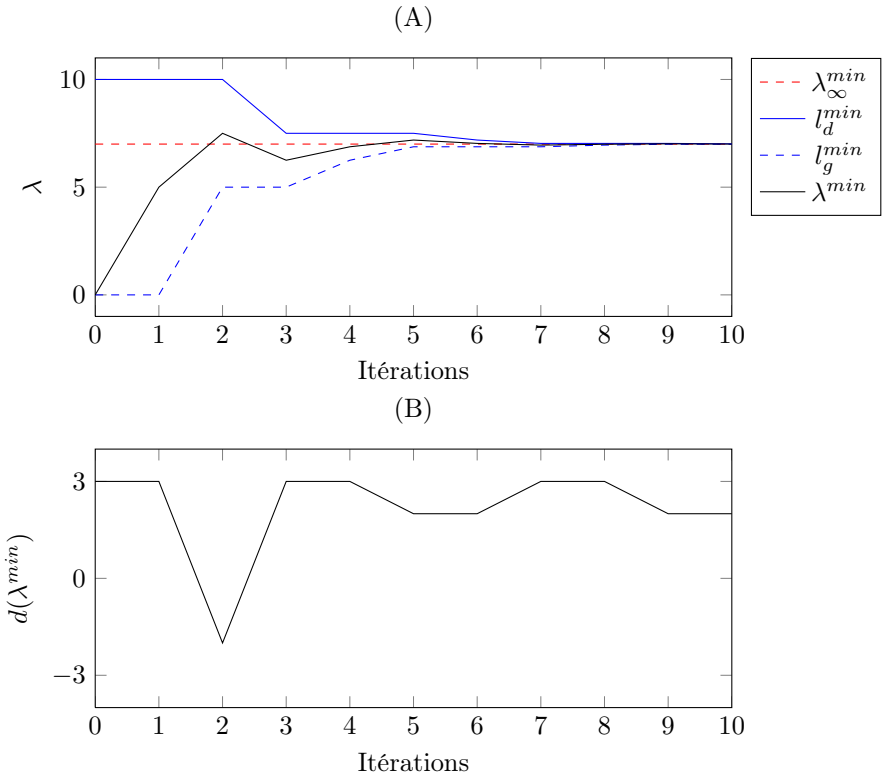



FIGURE *3.5 – Exemple du comportement de la recherche par dichotomie sur la borne inférieure* $\lambda^{min}$ *pour* $\lambda_\infty^{min} = 7$ *avec* $e = 3$.

Nous notons $\Lambda_r$ le nouvel ensemble de $\lambda$, $r$ signifiant *réduit*. Pour la recherche de $\Lambda_r$, nous utilisons la notion de voisinage et donc le paramètre $e$. Il est possible de dissocier le paramètre $e$ utilisé pour trouver $\Lambda_r$ du paramètre $e$ utilisé pour construire $\widehat{\mathcal{G}}$ à partir des solutions du Graphical lasso. Pour cela nous notons $e_r$ le paramètre $e$ utilisé pour trouver $\Lambda_r$. Pour la variante ABiGlasso

avec intervalle réduit, le schéma 3.2a de l'étape d'initialisation devient le schéma 3.6.

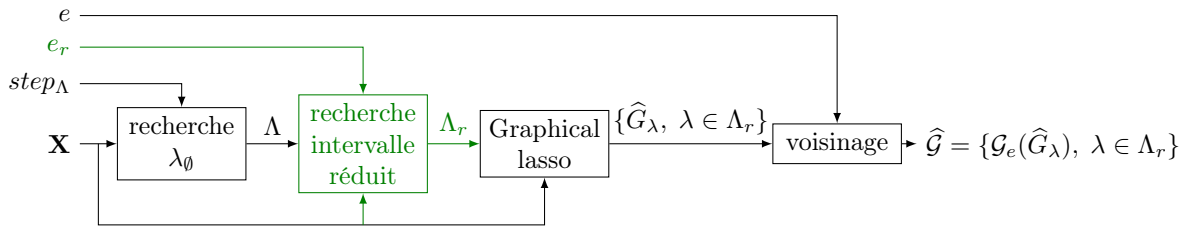



FIGURE *3.6 – Schéma de l'étape d'initialisation pour la méthode ABiGlasso avec intervalle réduit.*

La figure 3.7 illustre le fonctionnement de la variante ABiGlasso avec intervalle réduit sur le processus utilisé pour illustrer le fonctionnement de la méthode ABiGlasso de base. Sur cet exemple, pour les paramètres $e_r = 3$, $e = 3$ et $step_\Lambda = 0.1$, la méthode a pour solution le graphe attendu.

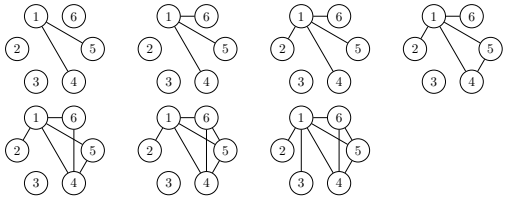
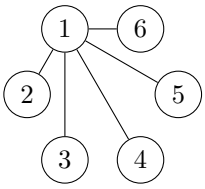
| $\mathbf{X}$ | cf figure 3.3 |
|---|---|
| $\Lambda = \{0, 0.1, 0.2, \ldots, 14.3\}$ | |
| $\Lambda_r = \{8.3, 8.4, 8.5, \ldots, 12.7\}$ | |
| $\{\widehat{G}_\lambda, \lambda \in \Lambda_r\}$ |  |
| $\lvert\widehat{\mathcal{G}}\rvert = \lvert\{\mathcal{G}_3(\widehat{G}_\lambda), \lambda \in \Lambda_r\}\rvert = 2335$ | |
| $\widehat{G}$ |  |

FIGURE *3.7 – Illustration des sorties des différentes étapes de la méthode ABiGlasso avec intervalle réduit sur le processus synthétique utilisé dans l'exemple figure 3.3. Sur ce processus, la méthode ABiGlasso avec intervalle réduit avec pour paramètres $e_r = 3$, $e = 3$ et $step_\Lambda = 0.1$ donne le résultat attendu.*

Cette méthode permet de réduire le nombre de graphes à comparer mais en générant d'autres sous-ensembles de graphes (en utilisant le voisinage), elle reste donc une alternative coûteuse en temps de calcul.

Information Toolbox : cette variante est implémentée dans la Toolbox. C'est la fonction *Method_ABiGlasso_RI*.

#### 3.1.2.3 Variante 2 : Comparaison des graphes dans le voisinage du graphe $\widehat{G}_\lambda$ le plus probable - ABiGlassoMax

Afin de limiter le nombre de graphes dans $\widehat{\mathcal{G}}$, dans cette variante, nous proposons de poser

$$\widehat{\mathcal{G}} = \mathcal{G}_e(\widehat{G}_\lambda^{max}) \text{ avec } \widehat{G}_\lambda^{max} = \underset{G \in \{\widehat{G}_\lambda, \lambda \in \Lambda\}}{\text{argmax}} \; z(G|\mathbf{X}).$$

Pour cela, il suffit de calculer le score $z$ pour l'ensemble des graphes $\widehat{G}_\lambda$ avec $\lambda \in \Lambda$ et de ne générer que le voisinage du graphe ayant le score le plus élevé. Nous appelons cette variante ABiGlassoMax pour *Asymptotic Bayesian method initialized by Graphical lasso maximization*. Le schéma 3.2a de l'étape d'initialisation devient le schéma 3.8.

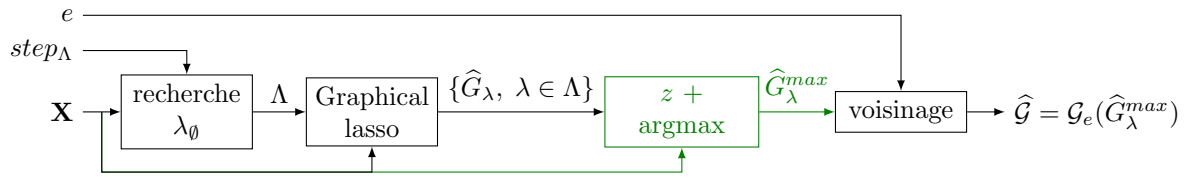



FIGURE *3.8 – Schéma de l'étape d'initialisation pour la méthode ABiGlassoMax.*

La figure 3.9 illustre le fonctionnement de la variante ABiGlassoMax sur le même processus que celui utilisé pour illustrer le fonctionnement de la méthode de base et de la variante avec intervalle réduit. Pour les paramètres $e = 3$ et $step_\Lambda = 0.1$, cette méthode donne la solution attendue.

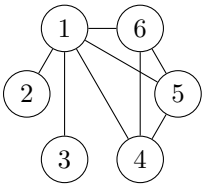
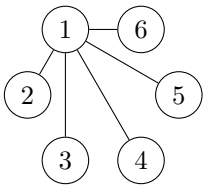
| | |
|---|---|
| $\mathbf{X}$ | cf figure 3.3 |
| $\Lambda = \{\lambda = 5.5\}$ | |
| $\widehat{G}_\lambda^{max}$ |  |
| $\lvert\widehat{\mathcal{G}}\rvert = \lvert\mathcal{G}_3(\widehat{G}_\lambda^{max})\rvert = 576$ | |
| $\widehat{G}$ |  |



FIGURE *3.9 – Illustration des sorties des différentes étapes de la méthode ABiGlassoMax sur le processus synthétique utilisé dans l'exemple figure 3.3. Sur ce processus, la méthode ABiGlassoMax avec pour paramètres $e = 3$ et $step_\Lambda = 0.1$ donne le résultat attendu.*

Le nombre de scores à calculer est plus faible que pour la méthode ABiGlasso avec réduction d'intervalle car nous devons uniquement calculer un score par élément de $\{\widehat{G}_\lambda, \lambda \in \Lambda\}$ et pour les $\gamma(e, p)$ graphes du voisinage de $\widehat{G}_\lambda^{max}$.

Dans le cas de non convergence du Graphical lasso, $\widehat{G}_\lambda^{max}$ peut s'avérer éloigné de la solution attendue, c'est-à-dire que la distance de Hamming entre les deux graphes est élevée. Pour compenser ce phénomène, il faut comparer les graphes d'un voisinage avec $e$ grand. Par exemple, pour le processus utilisé figure 3.9, la distance de Hamming entre $\widehat{G}_\lambda^{max}$ et la solution attendue vaut 3, on est donc obligé de parcourir un voisinage avec au moins $e = 3$, c'est-à-dire de comparer 576 graphes pour être sûr d'avoir la solution attendue.

Cette méthode permet de réduire la taille de $\widehat{\mathcal{G}}$ en n'explorant qu'un seul voisinage mais pour bien estimer la structure d'indépendance conditionnelle du processus étudié, il peut être nécessaire d'explorer un voisinage de paramètre $e$ grand.

Information Toolbox : cette variante est implémentée dans la Toolbox. C'est la fonction *Method_ABiGlasso_Max*.

#### 3.1.2.4 Variante 3 : Comparaison itérative de voisinages - ABiGlassoMaxLoop

L'objectif de cette variante est de comparer de manière itérative les graphes de voisinages de paramètre $e$ faible plutôt que d'explorer un voisinage avec $e$ grand.

Supposons que nous étudiions un processus à $p = 6$ variables et que le graphe solution du Graphical lasso qui maximise le score $z$ sur $\Lambda$ ait quatre arêtes de différence avec la structure de graphe attendue. Si nous utilisons la méthode ABiGlassoMax avec $e = 3$, la solution aura au mieux une arête de différence avec la structure de graphe attendue et jusqu'à 576 graphes $\widehat{G}_\lambda$ auront été parcourus lors de l'étape de comparaison. Si nous procédons de manière itérative avec $e = 1$, la solution obtenue sera la structure de graphe attendue et au maximum $4 \times 16 = 64$ graphes auront été parcourus. Ceci n'est qu'un cas particulier mais il permet de mettre en lumière la raison pour laquelle nous avons mis en place cette variante.

Pour cette variante, l'étape d'initialisation est la même que pour la variante ABiGlassoMax et c'est l'étape de comparaison qui est modifiée. Nous considérons que la méthode a convergé quand $\widehat{G}(i) = \widehat{G}(i-1)$. Nous appelons cette variante ABiGlassoMaxLoop car nous procédons de manière itérative à partir de l'initialisation par maximisation du score des solutions du Graphical lasso.

Le schéma de l'étape de comparaison 3.2b devient le schéma 3.10.

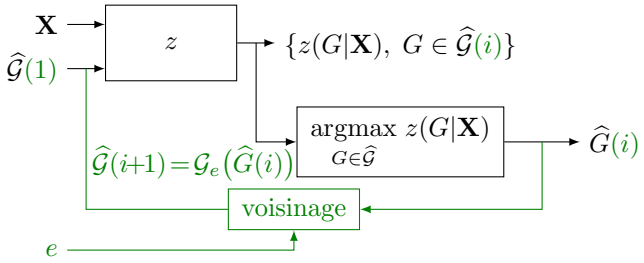



FIGURE *3.10 – Schéma de l'étape de comparaison pour la méthode ABiGlassoMaxLoop.*

La figure 3.11 illustre le fonctionnement de la variante ABiGlassoMaxLoop sur le processus synthétique utilisé pour les autres variantes. Pour $e = 1$ et $step_\Lambda = 0.1$, l'algorithme converge vers la solution attendue en 4 itérations.

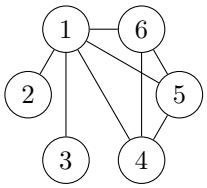
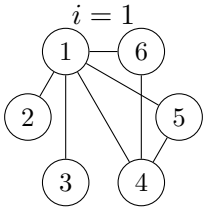
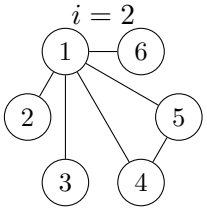
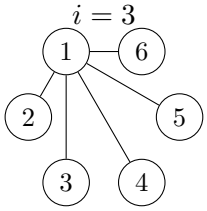
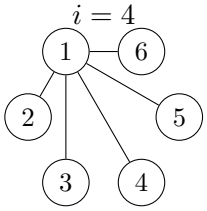
| $\mathbf{X}$ | cf figure 3.3 | | | |
|---|---|---|---|---|
| $\Lambda = \{\lambda = 5.5\}$ | | | | |
| $\widehat{G}_\lambda^{max}$ |  | | | |
| $\widehat{G}(i)$ | $i=1$  | $i=2$  | $i=3$  | $i=4$  |

FIGURE *3.11 – Illustration des sorties des différentes étapes de la méthode ABiGlassoMaxLoop sur le processus synthétique utilisé dans l'exemple figure 3.3. Sur ce processus, la méthode ABiGlassoMaxLoop avec pour paramètres* $e = 1$ *et* $step_\Lambda = 0.1$ *donne le résultat attendu après 4 itérations.*

Cette variante a pour avantage de parcourir moins de graphes que la variante ABiGlassoMax pour accéder à un graphe qui diffère de quelques arêtes $\widehat{G}_\lambda^{max}$. Cependant, elle est sensible aux maxima locaux : supposons que le graphe attendu $\widehat{G}$ ait plusieurs arêtes de différence avec $\widehat{G}_\lambda^{max}$, il est possible que $\widehat{G}(1) = \underset{G \in \widehat{\mathcal{G}}(2)}{\text{argmax}}\, z(G|\mathbf{X})$ et même si $z(\widehat{G}|\mathbf{X}) > z(\widehat{G}(1)|\mathbf{X})$, l'algorithme ne va pas converger vers $\widehat{G}$ mais vers $\widehat{G}(1)$.

Information Toolbox : cette variante est implémentée dans la Toolbox. C'est la fonction *Method_ABiGlasso_MaxLoop*.

### 3.1.3 Discussion

Nous avons introduit, dans les paragraphes précédents, notre méthode ABiGlasso et ses variantes. Nous avons illustré leurs fonctionnements sur un exemple pour des valeurs données des paramètres d'entrée des méthodes. Or pour d'autres exemples et d'autres valeurs des paramètres d'entrée, les performances des méthodes peuvent varier (nous avons vu que pour $e = 2$, la méthode ABiGlassoMax ne donne pas la solution attendue). Pour mieux comprendre ces méthodes, dans la suite de ce chapitre, nous allons dans un premier temps faire une étude théorique sur le temps de calcul de chacune de ces méthodes puis nous allons faire une étude empirique de l'influence des paramètres d'entrée sur le temps de calcul et sur la qualité de la solution proposée à partir de processus synthétiques dont la structure d'indépendance conditionnelle est connue.

## 3.2 Temps de calcul - Estimation et optimisation

Comme nous souhaitons une méthode dont le temps de calcul est modéré, cette partie donne quelques clés pour estimer le temps de calcul des différentes variantes.

### 3.2.1 Estimation des temps de calcul

Commençons par l'estimation du temps de comparaison des graphes d'un voisinage de paramètre $e$ pour un processus à $p$ variables. Nous notons ce temps $t_{comp}(e,p)$. La comparaison d'un voisinage comporte deux étapes, le calcul des scores de tous les graphes du voisinage et l'extraction du graphe le plus probable. Le graphe le plus probable étant obtenu en choisissant le graphe avec le score le plus élevé, cette étape nécessite un temps de calcul négligeable devant le calcul des scores $z$. Concernant le temps de calcul de $z$, la partie prépondérante est le calcul de $V(G)$. Le temps de calcul de l'étape de comparaison peut donc être assimilé au produit entre le temps de calcul du volume $V(G)$ et le nombre de graphes à comparer $\gamma(e,p)$ :

$$t_{comp}(e,p) \simeq t_{V(G)} \times \gamma(e,p) \tag{3.6}$$

$$\simeq t_{V(G)} \times \sum_{i=0}^{e} \frac{\frac{p(p-1)}{2}!}{\left(\frac{p(p-1)}{2}-i\right)!\; i!} \tag{3.7}$$

Toutes les méthodes ABiGlasso commencent par l'étape de recherche de $\Lambda$. Le temps de calcul de cette étape est noté : $t_\Lambda(\mathbf{X}, step_\Lambda)$. Cette étape dépend des données, il n'est donc pas possible d'estimer théoriquement son temps de calcul. Cependant, pour un processus $\mathbf{X}$ donné, plus $step_\Lambda$ est faible, plus le temps de calcul sera élevé et plus $step_\Lambda$ sera important plus le temps de calcul sera faible. En effet, ce paramètre permet de contrôler le moment où la convergence est atteinte. Pour l'ensemble des méthodes ABiGlasso nous avons aussi besoin de connaître $|\Lambda| = \lfloor \frac{\lambda_\emptyset}{step_\Lambda} \rfloor + 1$ ce qui n'est pas possible car $\lambda_\emptyset$ dépend également des données.

Pour la méthode ABiGlasso de base :

$$t_{base} \leq t_\Lambda(\mathbf{X}, step_\Lambda) + |\Lambda| \times (t_{Gl} + t_{comp}(e,p))$$

Nous pouvons uniquement majorer le temps de calcul car nous avons vu que plusieurs $\lambda \in \Lambda$ peuvent donner le même graphe et que un même graphe peut être contenu dans les voisinages de graphes distincts.

Pour la variante ABiGlasso avec réduction d'intervalle :

$$t_r \leq t_\Lambda(\mathbf{X}, step_\Lambda) + 2n_r^{it} \times (t_{Gl} + t_{comp}(e_r,p)) + |\Lambda_r| \times (t_{Gl} + t_{comp}(e,p))$$

où $n_r^{it}$ est le nombre d'itérations avant convergence de la recherche par dichotomie de $\Lambda_r$. Nous savons déjà que $t_\Lambda(\mathbf{X}, step_\Lambda)$ et $|\Lambda|$ ne sont pas accessibles car ils dépendent des données mais $n_r^{it}$ et $|\Lambda_r| = \left(\lceil \frac{\lambda_\infty^{max}}{step_\Lambda} \rceil - \lfloor \frac{\lambda_\infty^{min}}{step_\Lambda} \rfloor\right) + 1$ dépendent également des données et ne sont donc pas accessibles.

Pour la variante ABiGlassoMax :

$$t_{max} \leq t_\Lambda(\mathbf{X}, step_\Lambda) + |\Lambda| \times t_{Gl} + t_{comp}(e,p)$$

Pour la variante ABiGlassoMaxLoop :

$$t_{loop} \leq t_\Lambda(\mathbf{X}, step_\Lambda) + |\Lambda| \times t_{Gl} + t_{comp}(e,p) \times n^{it}$$

où $n^{it}$ est le nombre d'itérations avant convergence de l'étape de comparaison. Pour limiter le temps de calcul, il est possible d'introduire une alternative à la convergence et autoriser un nombre maximal d'itérations.

En fixant les paramètres d'entrée, la variante ABiGlassoMax est plus rapide que la méthode ABiGlasso de base puisqu'un seul voisinage est exploré au lieu de l'union des $|\Lambda|$ voisinages. La variante ABiGlassoMaxLoop, pour un $e$ donné est plus coûteuse en temps de calcul que ABiGlassoMax mais si le nombre d'itérations pour atteindre la convergence est faible, cette variante est plus rapide que la méthode ABiGlasso de base. Il est difficile de positionner la variante ABiGlasso avec réduction d'intervalle car son temps de calcul dépend de plusieurs paramètres non présents dans les autres variantes.

Pour savoir quelle variante est la plus pertinente, il faut étudier leurs performances en terme d'estimation de la structure d'indépendance conditionnelle. Mais avant, nous étudions la possibilité d'optimiser le temps de calcul de ces variantes en réduisant le temps de calcul de $V(G)$.

### 3.2.2 Optimisation du temps de calcul

La partie prépondérante dans le temps de calcul est l'estimation de $V(G)$. Pour optimiser le temps de calcul, il faut optimiser le calcul des $V(G)$.

L'estimation de $V(G)$ se fait en utilisant une méthode de Monte Carlo.

Rappel : Calcul de $V(G)$ (cf [MB06a])

$V(G)$ est le volume de $f(\mathcal{M}_0^+(G))$ où $\mathcal{M}_0^+(G)$ est l'ensemble des matrices symétriques définies positives ayant pour valeur 0 quand $(i,j) \notin E$ et $f$ est la transformation qui permet de passer de la matrice de précision à la matrice des corrélations partielles. Nous notons $\mathcal{M}^+$ l'ensemble des matrices symétriques définies positives. Comme les valeurs des éléments hors diagonale de la matrice des corrélations partielles sont dans l'intervalle $[-1,1]$, on peut majorer $V(G)$ par $2^{p(p-1)}$ (nombre d'éléments hors diagonale) où $p$ est le nombre de nœuds du graphe $G$. Si on pose $V_p = 2^{p(p-1)/2}$, on peut estimer la fraction $k(G) = V(G)/V_p$ de la façon suivante :

- on choisit $L$ échantillons de $\boldsymbol{\pi}_{|E|}$ uniformément dans l'hypercube $[-1,1]^{|E|}$
- on approche $k(G)$ par $\frac{1}{L}\sum_{l=1}^{L} \mathbb{1}_{f(\mathcal{M}^+)}(\boldsymbol{\pi}^{[l]})$, où $\mathbb{1}_{f(\mathcal{M}^+)}(\boldsymbol{\pi}^{[l]}) = 1$ si $\boldsymbol{\pi}^{[l]} \in f(\mathcal{M}^+)$ et 0 sinon.

$\boldsymbol{\pi} \in f(\mathcal{M}^+)$ si et seulement si $2I - \Pi \in \mathcal{M}^+$. Nous rappelons que $\boldsymbol{\pi}$ est la version vectorisée de la matrice $\Pi$.

#### 3.2.2.1 Sauvegarde des $V(G)$ déjà calculés

Sauvegarder les $V(G)$ déjà calculés permet de ne pas avoir à les recalculer si ils ont déjà été calculés précédemment soit dans une itération précédente pour la méthode avec comparaison successive de voisinages, soit lors de l'étude d'un autre processus. Le problème est qu'il y a $2^{p(p-1)/2}$ graphes possibles pour un processus à $p$ variables et qu'il n'est pas possible, du moins sous Matlab, de générer un vecteur contenant tous les $V(G)$ pour $p > 8$. Si on ne souhaite sauvegarder que les $V(G)$ calculés en utilisant par exemple un dictionnaire, le temps de calcul pour vérifier si un $V(G)$ a déjà été calculé augmente avec le nombre de $V(G)$ calculés et peut s'avérer plus coûteux en temps de calcul que de l'estimer.

*Note :* pour $p = 6$, nous avons pu stocker les estimées de tous les $V(G)$ possibles.

#### 3.2.2.2 Réduction du nombre de $V(G)$ à calculer

Une autre méthode que nous avons envisagée est de n'avoir qu'un seul $V(G)$ pour un ensemble de graphes isomorphes.

**Définition 3.2.1.** ***Graphes isomorphes :***

*Soient deux graphes $G$ et $H$, et leurs matrices d'adjacence respectivement $A_G$ et $A_H$.*

*Soit perm l'ensemble des fonctions de permutations telles que $N = h(M)$ avec $h \in perm$ s'obtienne en permutant des lignes de $M$ et en permutant de la même façon les colonnes associées. Par exemple si $h$ permute les lignes 2 et 4 (et donc les colonnes 2 et 4) de $M$ :*

$$N = \begin{pmatrix} m_{11} & m_{14} & m_{13} & m_{12} \\ m_{41} & m_{44} & m_{43} & m_{42} \\ m_{31} & m_{34} & m_{33} & m_{32} \\ m_{21} & m_{24} & m_{23} & m_{22} \end{pmatrix}$$

*$G$ et $H$ sont isomorphes si et seulement si $A_H = h(A_G)$, $h \in perm$.*

La figure 3.12 illustre les différents isomorphismes pour les graphes à $p = 3$ nœuds. En ne considérant que les graphes non-isomorphes, le nombre de graphes est réduit de 8 à 4. Pour les graphes à $p = 6$ nœuds, il y a 156 graphes non-isomorphes pour 32 768 graphes possibles.

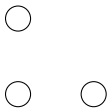
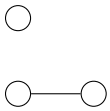
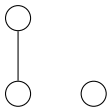
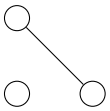
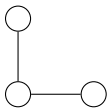
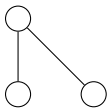
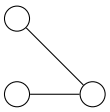
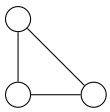
| graphe référence | graphes isomorphes |
|---|---|
|  $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ | - |
|  $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ |  $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ permutation 2 - 3  $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ permutation 1 - 3 |
|  $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ |  $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ permutations 1 - 2 et 2 - 3  $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ permutation 1 - 2 |
|  $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ | - |

FIGURE *3.12 – Isomorphisme pour les graphes à $p = 3$ nœuds. Les 8 graphes possibles avec 3 nœuds sont regroupés en 4 isomorphismes.*

Nous allons dans un premier temps démontrer que si deux graphes $G$ et $H$ sont isomorphes alors $V(G) = V(H)$.

**Démonstration de $V(G) = V(H)$ si $G$ et $H$ sont isomorphes**

Soient $\mathcal{M}_0(G)$ l'ensemble des matrices ayant des 0 là où il n'y a pas d'arêtes dans le graphe $G$, $\mathcal{M}^+$ l'ensemble des matrices définies positives et $\mathcal{M}_0^+(G) = \mathcal{M}^+ \cap \mathcal{M}_0(G)$.

Soit $f$ la fonction qui permet de passer de la matrice de précision à la matrice des corrélations partielles.

$V(G)$ correspond au volume de $f(\mathcal{M}_0^+(G))$. Pour le calculer empiriquement, des matrices $\Pi_G$ sont générées, elles doivent appartenir à $\mathcal{M}_0^+(G)$ et avoir leurs éléments compris entre -1 et 1. Ces matrices appartiennent à $f(\mathcal{M}_0^+(G))$ si et seulement si $2I - \Pi_G \in \mathcal{M}^+$, il faut donc vérifier que les valeurs propres de $2I - \Pi_G$ sont non nulles.

Dire que $V(G) = V(H)$ revient à dire que pour toute matrice $\Pi_G$, il existe une matrice $\Pi_H$ telle que si $2I - \Pi_G \in \mathcal{M}^+$ alors $2I - \Pi_H \in \mathcal{M}^+$, ce qui équivaut à ce que les valeurs propres de $2I-\Pi_H$ soient les mêmes que celles de $2I-\Pi_G$ donc que $\det((2-\lambda)I-\Pi_H) = \det((2-\lambda)I-\Pi_G)$.

Si $H$ et $G$ sont isomorphes, $A_H = h(A_G)$, $h \in perm$ et il existe une matrice $\Pi_H$ telle que $\Pi_H = h(\Pi_G)$.

Sachant que les éléments sur la diagonale de toute matrice $\Pi$ valent 1, la diagonale est invariante à toute permutation $h$. Sachant cela, il est évident que $(2-\lambda)I-\Pi_H = h((2-\lambda)I-\Pi_G)$.

$h$ étant toujours composée d'un nombre pair de permutations (une permutation entre deux lignes implique toujours une permutation entre les deux colonnes associées et réciproquement), $\det((2-\lambda)I - \Pi_H) = \det((2-\lambda)I - \Pi_G)$.

En conclusion, deux graphes isomorphes ont le même $V(G)$.

□

Cette approche prometteuse présente cependant un gros inconvénient : pour des graphes de grande dimension, il est très coûteux voire impossible de vérifier si deux graphes sont isomorphes. En effet, vérifier si deux graphes sont isomorphes est un problème connu sous le nom de *problème d'isomorphisme de graphe*, problème dont la complexité n'a toujours pas été établie [AT05]. Utiliser les propriétés d'isomorphisme pour le calcul de $V(G)$ ne fait donc que déplacer, voire complexifier, le problème à l'origine du coût de calcul.

#### 3.2.2.3 Conclusion sur l'optimisation du calcul de $V(G)$

Nous n'avons pas trouvé d'alternative à la méthode actuellement implémentée pour le calcul de $V(G)$.

## 3.3 Performances sur des processus simulés

Pour évaluer les performances des variantes de notre méthode ABiGlasso, nous utilisions la procédure d'évaluation introduite dans le chapitre 2. Dans un premier temps, nous présentons les processus tests que nous avons choisi d'utiliser pour notre procédure d'évaluation. Ensuite, nous étudions le comportement des méthodes en fonction du nombre d'observations des processus. Dans un dernier temps, nous étudions l'influence des paramètres d'entrée sur le temps de calcul et sur les performances en terme de qualité de la solution proposée.

### 3.3.1 Choix des processus tests

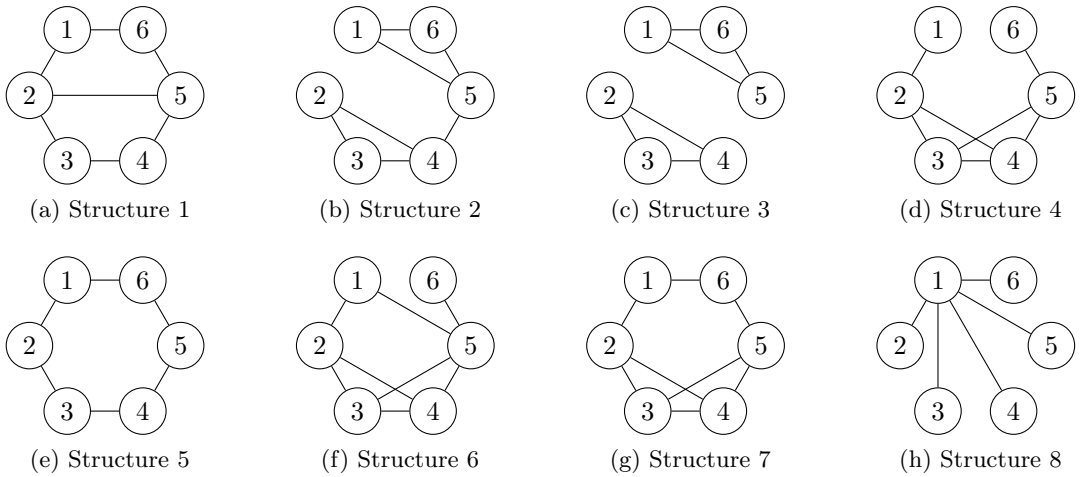
Pour des raisons de temps de calcul, nous utilisons des processus à $p = 6$ variables. Ce nombre de nœuds nous permet également de faire des simulations relativement exhaustives

#### 3.3.1.1 Structures

L'idéal est d'étudier les méthodes sur toutes les structures à $p = 6$ nœuds mais cela n'est pas possible pour des raisons de temps de calcul. Nous avons utilisé huit structures représentatives

des différents types de graphes. Ces structures sont présentées dans la figure 3.13. Elle ont les propriétés suivantes :

- certaines structures sont décomposables (3.13b,3.13c,3.13d), les autres non → pour comparer aux méthodes bayésiennes uniquement sur les graphes décomposables
- la structure (3.13c) est séparable → structure en deux morceaux
- les structures 3.13f et 3.13g sont non décomposables mais ne diffèrent de la structure décomposable 3.13d que d'une seule arête → pour étudier l'influence de la décomposabilité.
- la structure 3.13h est une structure en étoile → structure problématique pour la convergence du Graphical lasso.



(a) Structure 1 (b) Structure 2 (c) Structure 3 (d) Structure 4

(e) Structure 5 (f) Structure 6 (g) Structure 7 (h) Structure 8

FIGURE *3.13 – Différentes structures d'indépendance conditionnelle à 6 nœuds utilisées pour évaluer les performances des méthodes ABiGlasso. Ces structures représentent des configurations de graphes que nous souhaitons retrouver avec nos méthodes : graphes décomposables (3.13b, 3.13c, 3.13d), graphes non-décomposables (3.13a, 3.13e, 3.13f, 3.13g), graphes séparables (3.13c) et graphes présentant des groupes de nœuds fortement connectés (3.13d, 3.13f, 3.13g et 3.13h).*

#### 3.3.1.2 Matrices théoriques

Pour chacune des structures, nous souhaitons générer des matrices de covariance dont l'inverse respecte la structure choisie. Pour cela, nous utilisons la méthode présentée dans le chapitre 2. Nous choisissons $s = 0.2$ pour des raisons que nous allons voir dans la section sur le choix du nombre d'observations des processus simulés. Pour chaque structure, nous générons 10 matrices $\Sigma$ différentes.

#### 3.3.1.3 Nombre d'observations

A partir de chacune des $8 \times 10$ matrices de covariance $\Sigma$ générées selon les structures de graphes présentées dans la figure 3.13 et pour $n \in \{7, 12, 60, 600\}$ nous générons 10 processus. Au final, nous avons 800 processus pour chaque valeur de $n$.

La justification du choix des différentes valeurs de $n$ est le suivant :

- $n = 7$ : cas $n \simeq p$,
- $n = 12$ : $n$ légèrement plus important que $p$,
- $n = 60$ : $n > p$,
- $n = 600$ : $n >> p$.

De plus, comme nous avons choisi $s = 0.2$, nous sommes certains que les corrélations partielles non nulles des matrices générées précédemment sont significativement non nulles uniquement

pour le cas $n = 600$. En effet comme vu dans la section 2.1.2.2 (chapitre 2), avec un risque de faux négatif de 1% et $p = 6$, le seuil au-dessus duquel une corrélation partielle, en valeur absolue, est significativement non nulle vaut 0.134 pour $n = 600$ et 0.424 pour $n = 60$.

### 3.3.2 Influence du nombre d'observations

Nous étudions des processus ayant différents nombres d'observations $n$ dans l'objectif de savoir pour quels processus nos méthodes donnent des résultats pertinents et pour quels processus il est inutile d'appliquer l'une ou l'autre des méthodes car les résultats ne seront pas pertinents.

La figure 3.14 représente les histogrammes de la distance de Hamming entre les solutions proposées et les structures de graphes imposées aux processus pour la méthode ABiGlasso de base avec le paramètre $e = 5$ où $e$ est le paramètre de voisinage.

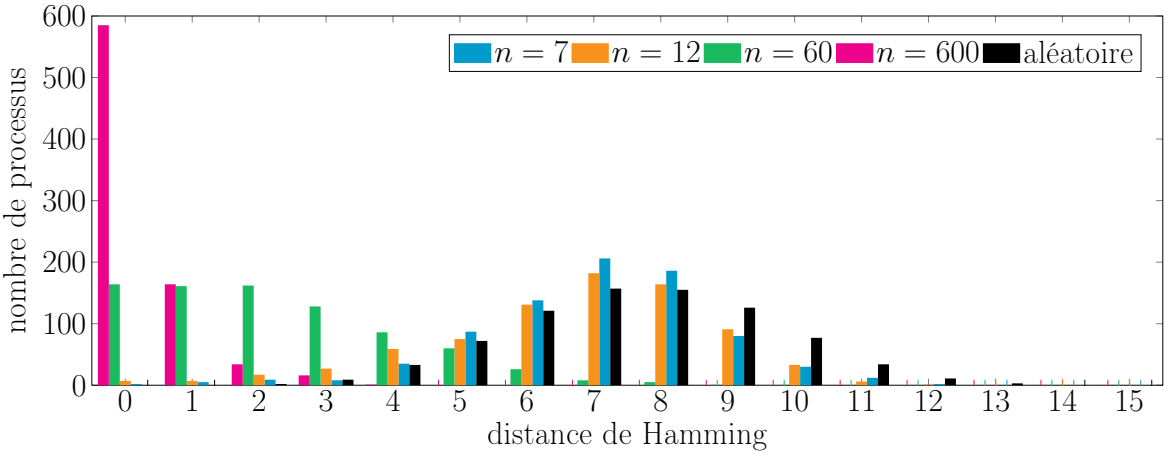



FIGURE *3.14 – Exemple des histogrammes de la distance de Hamming en fonction de n pour la méthode ABiGlasso de base avec e = 5.*

Pour $n = 7$ et $n = 12$ les résultats sont assimilables aux résultats obtenus quand le graphe solution est choisi aléatoirement parmi tous les graphes possibles. Pour $n = 60$, la solution obtenue est plus proche de la solution attendue, la distance de Hamming moyenne étant de 2.34 sur l'ensemble des 800 processus. Pour $n = 600$, les performances sont très bonnes avec une moyenne de la distance de Hamming de 0.44.

Plus généralement, le tableau 3.1 montre que quelques soient la méthode ABiGlasso et les paramètres d'entrée utilisés, les performances pour $n = 7$ et $n = 12$ sont mauvaises : en moyenne, la distance de Hamming est supérieure à 7.5 pour $n = 7$, ce qui correspond à un choix aléatoire, et supérieure à 6.5 pour $n = 12$ ce qui plus proche de l'aléatoire que d'une extraction parfaite (distance de Hamming nulle). Pour $n = 60$, la distance de Hamming est comprise en moyenne entre 2 et 3, ce qui est correct et pour $n = 600$ elle est, en moyenne, proche de 0 ce qui signifie que l'extraction donne de bons résultats.

Ces observations amènent aux conclusions suivantes :

- Si $n$ est trop proche de $p$, les solutions obtenues sont similaires à un choix aléatoire, il est donc préférable de ne pas utiliser ABiGlasso dans ces conditions.
- Pour $n > p$, les méthodes ABiGlasso permettent de donner un graphe proche, en terme de distance de Hamming, de la solution attendue, les méthodes peuvent donc être utilisées en ayant un regard critique sur la solution obtenue qui est porteuse d'information mais aussi d'une variance intrinsèque à l'approche d'estimation.
- Pour $n >> p$, les méthodes ABiGlasso sont performantes et la solution obtenue peut être exploitée en l'état.

Dans le chapitre suivant, nous verrons le comportement d'autres méthodes de la littérature pour ces différentes conditions.

### 3.3.3 Influence des paramètres d'entrée

#### 3.3.3.1 Influence sur la qualité de la solution obtenue

Le tableau 3.1 donne la distance de Hamming moyenne sur les 800 processus pour un $n$ donné pour les différentes méthodes ABiGlasso avec différentes valeurs de $e$ (paramètre de voisinage) et de $e_r$ (paramètre de voisinage pour la recherche d'intervalle). Nous avons vu à la section précédente que pour $n = 7$ et $n = 12$, les résultats ne sont pas plus intéressants que des résultats obtenus aléatoirement, nous focalisons donc notre étude sur les résultats obtenus pour $n = 60$ et $n = 600$.

**Paramètre $e$**

Ce paramètre définit la taille du voisinage exploré lors de l'étape de comparaison : le voisinage d'un graphe $G$ de paramètre $e$ contient tous les graphes qui diffèrent de $G$ d'au plus $e$ arêtes.

D'après le tableau 3.1, pour toutes les méthodes, à l'exception de ABiGlassoMaxLoop pour $n = 60$, la valeur de $e$ donnant le meilleur résultat est la valeur de $e$ la plus élevée. En effet c'est celle qui permet de s'éloigner au plus des solutions données par le Graphical lasso qui peuvent être assez éloignées en terme de distance de Hamming de la structure de graphe attendue à cause des problèmes de convergence.

**Paramètre $e_r$ pour ABiGlasso avec réduction d'intervalle**

Ce paramètre définit le nombre d'arêtes de différence maximal autorisé entre une solution du Graphical lasso et son voisin le plus probable lors de l'étape de réduction d'intervalle de la variante avec réduction d'intervalle.

Pour cette méthode, d'après le tableau 3.1, le choix du $e_r$ le plus pertinent n'est pas évident car les résultats dépendent majoritairement du choix de $e$. Cependant, pour $e$ fixé, les performances sont meilleures pour $e_r = 3$. Cela concorde avec le fait que les observations à l'origine de la création de la méthode ABiGlasso avec réduction d'intervalle ont été faites avec $e = 3$.

**Paramètre $step_\Lambda$**

Ce paramètre est le pas entre deux valeurs du paramètre de pénalisation $\lambda$ utilisées pour générer un graphe à partir du Graphical lasso.

Il influence $t_\Lambda$ et le nombre de valeurs de $\Lambda$ ($\lfloor \frac{\lambda_\emptyset}{step_\Lambda} \rfloor + 1$) et de $\Lambda_r$ ($\left( \lceil \frac{\lambda^{max}}{step_\Lambda} \rceil - \lfloor \frac{\lambda^{min}}{step_\Lambda} \rfloor \right) + 1$). Son influence n'est pas présentée dans le tableau 3.1 où il a été fixé à 0.1. Plus $step_\Lambda$ diminue, plus le nombre de paramètres de pénalisation à utiliser augmente, plus le nombre de graphes différents obtenus à partir du Graphical lasso augmente et donc plus les chances de parcourir le graphe le plus représentatif du processus sont importantes. Cependant, plus $step_\Lambda$ est petit, plus la complexité augmente.

#### 3.3.3.2 Influence sur le temps de calcul

Les mesures de temps de calcul ont été faites sur un processeur 2.66 GHz Intel Core i7.

**Temps de recherche de l'intervalle $\Lambda$, $t_\Lambda$**

$t_\Lambda$ dépend du processus considéré et de $step_\Lambda$.

La figure 3.15 représente l'histogramme des valeurs prises par $t_\Lambda$ sur les $800 \times 4$ processus simulés pour différentes valeurs du pas $step_\Lambda$. Plus le pas est grand, plus le temps de calcul est

| | ABiGlasso | ABiGlasso RI* | | | | ABiGlasso Max | ABiGlasso MaxLoop |
|---|---|---|---|---|---|---|---|
| $e$ \ $e_r$ | x | 1 | 2 | 3 | 4 | x | x |
| 1 | 7.30±1.36 | 7.85±1.45 | 8.01±1.42 | 8.07±1.40 | 8.05±1.41 | **7.46**±1.59 | **7.40**±1.52 |
| 2 | 7.19±1.40 | 7.67±1.51 | 7.79±1.52 | 7.84±1.52 | 7.81±1.53 | 7.64±1.55 | 7.57±1.57 |
| 3 | 7.04±1.53 | 7.59±1.58 | 7.73±1.54 | 7.76±1.55 | 7.75±1.56 | 7.67±1.55 | 7.67±1.59 |
| 4 | 6.97±1.66 | **7.51**±1.67 | 7.71±1.57 | 7.74±1.58 | 7.72±1.59 | 7.68±1.56 | 7.54±1.63 |
| 5 | **6.95**±1.75 | **7.51**±1.68 | 7.70±1.58 | 7.73±1.59 | 7.72±1.60 | 7.71±1.56 | 7.59±1.65 |

* ABiGlasso RI : ABiGlasso avec réduction d'intervalle.

(a) $n = 7$

| | ABiGlasso | ABiGlasso RI* | | | | ABiGlasso Max | ABiGlasso MaxLoop |
|---|---|---|---|---|---|---|---|
| $e$ \ $e_r$ | x | 1 | 2 | 3 | 4 | x | x |
| 1 | 6.95±1.99 | 7.20±1.90 | 7.27±1.87 | 7.31±1.84 | 7.29±1.85 | **6.58**±1.82 | **6.67**±1.92 |
| 2 | 6.84±1.93 | 6.96±1.93 | 7.02±1.89 | 7.05±1.88 | 7.04±1.88 | 6.73±1.84 | 6.76±1.94 |
| 3 | 6.69±1.99 | 6.81±1.97 | 6.83±1.96 | 6.84±1.95 | 6.84±1.95 | 6.77±1.86 | 6.77±1.95 |
| 4 | **6.65**±2.02 | 6.75±1.98 | 6.78±1.97 | 6.78±1.96 | 6.78±1.97 | 6.79±1.93 | 6.75±1.98 |
| 5 | 6.67±2.02 | **6.74**±1.98 | 6.76±1.98 | 6.77±1.97 | 6.77±1.97 | 6.79±1.94 | 6.76±1.97 |

* ABiGlasso RI : ABiGlasso avec réduction d'intervalle.

(b) $n = 12$

| | ABiGlasso | ABiGlasso RI* | | | | ABiGlasso Max | ABiGlasso MaxLoop |
|---|---|---|---|---|---|---|---|
| $e$ \ $e_r$ | x | 1 | 2 | 3 | 4 | x | x |
| 1 | 2.99±1.95 | 2.82±1.92 | 2.81±1.92 | 2.78±1.91 | 2.80±1.94 | 2.92±1.91 | 2.17±1.76 |
| 2 | 2.62±1.90 | 2.45±1.87 | 2.43±1.88 | 2.41±1.89 | 2.43±1.90 | 2.55±1.83 | **2.16**±1.78 |
| 3 | 2.33±1.83 | 2.23±1.79 | 2.20±1.79 | **2.18**±1.78 | **2.18**±1.78 | 2.30±1.78 | 2.17±1.79 |
| 4 | 2.24±1.82 | 2.19±1.79 | 2.19±1.79 | **2.18**±1.80 | 2.19±1.79 | 2.22±1.81 | 2.18±1.80 |
| 5 | **2.21**±1.80 | 2.19±1.80 | 2.19±1.80 | **2.18**±1.80 | **2.18**±1.80 | **2.19**±1.79 | 2.17±1.79 |

* ABiGlasso RI : ABiGlasso avec réduction d'intervalle.

(c) $n = 60$

| | ABiGlasso | ABiGlasso RI* | | | | ABiGlasso Max | ABiGlasso MaxLoop |
|---|---|---|---|---|---|---|---|
| $e$ \ $e_r$ | x | 1 | 2 | 3 | 4 | x | x |
| 1 | 1.87±2.13 | 1.20±1.33 | 1.19±1.33 | 1.15±1.25 | 1.19±1.34 | 1.30±1.33 | 0.27±0.56 |
| 2 | 1.25±1.85 | 0.71±1.05 | 0.64±0.88 | 0.62±0.86 | 0.64±0.89 | 0.78±1.02 | **0.25**±0.54 |
| 3 | 0.72±1.16 | 0.43±0.83 | 0.38±0.66 | 0.34±0.63 | 0.36±0.66 | 0.45±0.77 | **0.25**±0.52 |
| 4 | 0.51±0.90 | 0.33±0.68 | 0.28±0.54 | **0.25**±0.53 | 0.27±0.54 | 0.30±0.60 | **0.25**±0.52 |
| 5 | **0.36**±0.67 | 0.28±0.60 | **0.25**±0.52 | **0.25**±0.52 | **0.25**±0.52 | **0.27**±0.55 | **0.25**±0.52 |

* ABiGlasso RI : ABiGlasso avec réduction d'intervalle.

(d) $n = 600$

TABLEAU *3.1 – Distance de Hamming (moyenne ± écart-type sur les* 800 *processus), en fonction de $n$, $e$ et $e_r$ entre la structure de graphe imposée et le graphe solution pour la méthode ABiGlasso et ses trois variantes. Ces tableaux contiennent les valeurs obtenues pour $step_\Lambda = 0.1$. Les valeurs en gras représentent les plus petites distances de Hamming obtenues pour une méthode et une valeur de $n$ données.*

faible. Comme ce pas contrôle la convergence de l'étape de recherche de $\lambda_\emptyset$, plus il est grand, plus la convergence est atteinte rapidement donc les résultats observés concordent avec les résultats attendus.

Il est important de noter que quelle que soit la valeur de $step_\Lambda$ utilisée, $t_\Lambda$ ne dépasse pas les 3.4 secondes, cette étape n'est donc pas contraignante en terme de temps de calcul.

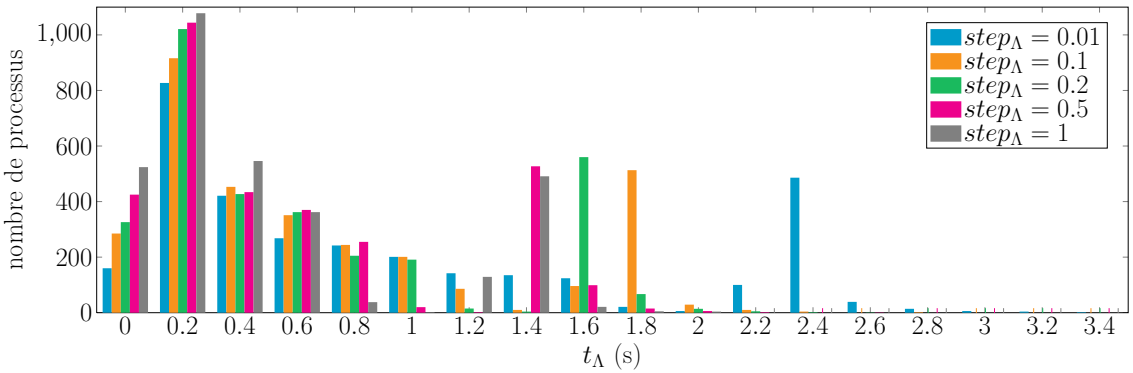



FIGURE *3.15 – Histogrammes des $t_\Lambda$ sur les $800 \times 4$ processus pour différentes valeurs de $step_\Lambda$. Ces temps sont indépendants des structures de graphe et du nombre d'observations.*

**Taille de $\Lambda$**

Le temps de calcul dépend de la taille de $\Lambda$ qui elle-même dépend de $step_\Lambda$ et $\lambda_\emptyset$.

$\lambda_\emptyset$ dépend principalement du processus et très peu du choix de $step_\Lambda$. En effet, pour un processus donné, l'écart type entre les $\lambda_\emptyset$ obtenus pour les 5 valeurs de $step_\Lambda$ vaut au plus 0.27 pour des valeurs de $\lambda_\emptyset$ allant de 0.63 à 20.55.

Comme les variations de $\lambda_\emptyset$ sont faibles pour un processus donné, si on fixe le processus, la taille de $\Lambda$ dépend uniquement de $step_\Lambda$ : par exemple, pour $step_\Lambda = 0.01$, $\Lambda$ a 100 fois plus de valeurs que pour $step_\Lambda = 1$.

**Temps d'exploration d'un voisinage $t_{comp}(e, p)$**

$t_{V(G)} \simeq 0.12$ secondes pour $p = 6$. Il ne dépend d'aucun paramètre d'entrée. Ce qui dépend des paramètres d'entrée est le nombre de fois qu'il faudra calculer $V(G)$.

Nous rappelons que $t_{comp}(e, p) \simeq t_{V(G)} \times \gamma(e, p)$ où $\gamma(e, p)$ est le nombre de graphes dans le voisinage de paramètre $e$ d'un graphe à $p$ nœuds. Ici $p = 6$ et nous testons les méthodes avec $e$ variant de 1 à 5. Le tableau 3.2 donne la valeur de $\gamma(e, 6)$ en fonction des $e$ étudiés.

| $e$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\gamma(e, 6)$ | 15 | 121 | 576 | 3306 | 9312 |
| $t_{comp}(e, 6)$ $(\simeq)$ | 1.8s | 14.52s | 1m09.12s | 6m36.72s | 18m37.44s |

TABLEAU *3.2 – Nombre $\gamma(e, 6)$ de graphes dans le voisinage d'un graphe à $p = 6$ nœuds en fonction du paramètre de voisinage $e$ et temps de calcul pour la comparaison des graphes dans ce voisinage, $t_{comp}(e, 6)$, aussi en fonction de $e$.*

Comme nous l'avons dit précédemment, la comparaison de graphes au sein de plusieurs voisinages de grandes tailles peut s'avérer très coûteuse en temps de calcul, même pour des processus à faible nombre de variables.

**Temps de calcul du Graphical lasso $t_{Gl}$**

Le temps d'exécution du Graphical lasso ne dépend pas du nombre d'observations ni de la structure de graphe mais il varie d'un processus à l'autre. Sur les $800 \times 4$ processus simulés, $t_{Gl}$ varie de 0.13 à 3.87 secondes avec une moyenne de 1.31 secondes. Si le nombre de fois où il faut appliquer le Graphical lasso est très élevé, cela peut engendrer des coûts de calcul important.

**Nombre d'itérations pour obtenir $\Lambda_r$**

Le nombre d'itérations pour la recherche de $\Lambda_r$ dépend de $\lambda_\emptyset$, $e_r$ et $n$. Plus $\lambda_\emptyset$ est élevé, plus l'espace dans lequel chercher $\Lambda_r$ est grand et plus le nombre d'itérations pour atteindre la convergence est important. Comme la convergence de l'algorithme de recherche est atteinte quand $l_d - l_g < 0.02$, et que la valeur maximale de $\lambda_\emptyset$ vaut 20.55 pour nos processus, le nombre d'itérations maximal est de 12.

$$
\begin{aligned}
l_d(i) - l_g(i) &= \frac{\lambda_\emptyset}{2^{(i-2)}} \ \forall i \geq 2 \\
\text{d'où } l_d(i+1) - l_g(i+1) < 0.02 &\Leftrightarrow \frac{\lambda_\emptyset}{2^{(i-1)}} < 0.02 \\
&\Leftrightarrow \lambda_\emptyset < 0.02 \times 2^{(i-1)} \\
&\Leftrightarrow \frac{\log(\lambda_\emptyset) - \log(0.02)}{\log(2)} + 1 < i \\
\text{Si } \lambda_\emptyset = 20.55, 11 &< i
\end{aligned}
$$

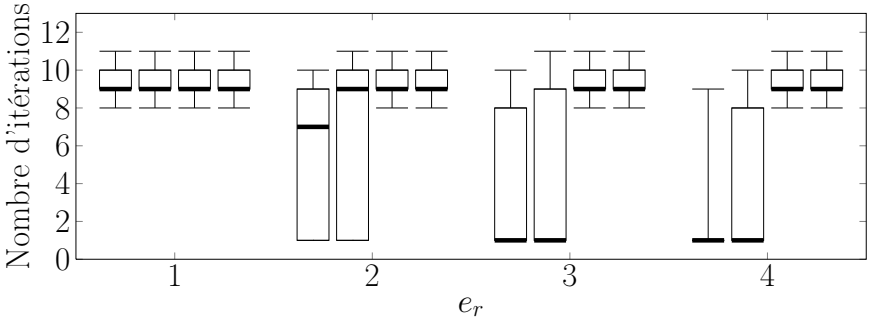



FIGURE *3.16 – Nombre d'itérations pour la recherche de $\Lambda_r$ avec différentes valeurs de $e_r$ et différentes valeurs de $n$. Pour une valeur de $e_r$, de gauche à droite, les boîtes à moustaches correspondent à $n = 7$, $n = 12$, $n = 60$ et $n = 600$.*

La figure 3.16 montre le nombre d'itérations sur les 800 processus pour différentes valeurs de $n$ et de $e_r$. On remarque que pour certaines configurations, la convergence est atteinte en une itération. Cela correspond au cas où $\Lambda_r = \Lambda : d(0) < e_r$ et $d(\lambda_\emptyset) > -e_r$. Ce cas est principalement rencontré pour des faibles valeurs de $n$ et des valeurs élevées de $e_r$. Comme nous avons vu que les cas $n = 7$ et $n = 12$ n'étaient pas des cas pertinents pour l'extraction de la structure d'indépendance conditionnelle, nous ne nous attardons pas sur ces configurations.

Notons que dans le cas de 12 itérations, il faut parcourir $2 \times 12$ voisinages de paramètres $e_r$ pour obtenir $\Lambda_r$. Pour que la variante ABiGlasso avec réduction d'intervalle soit avantageuse en terme de temps de calcul, il faut au moins que $|\widehat{G}_\lambda, \lambda \in \Lambda_r| < |\widehat{G}_\lambda, \lambda \in \Lambda| - 24$. Cette condition n'est pas suffisante car l'étape de comparaison se fait sur l'intersection des voisinages de $G \in \widehat{G}_\lambda, \lambda \in \Lambda$.

Comme pour $\Lambda$, $\Lambda_r$ dépend de $step_\Lambda$ : pour $step_\Lambda = 0.01$, $\Lambda_r$ a 100 fois plus d'éléments que pour $step_\Lambda = 1$.

**Nombre d'itérations pour la méthode ABiGlassoMaxLoop**

Le nombre d'itérations pour la convergence de la méthode ABiGlassoMaxLoop dépend de $e$ et de $n$ : plus $n$ est grand, plus le nombre d'itérations tend vers 1 (le minimum possible), de

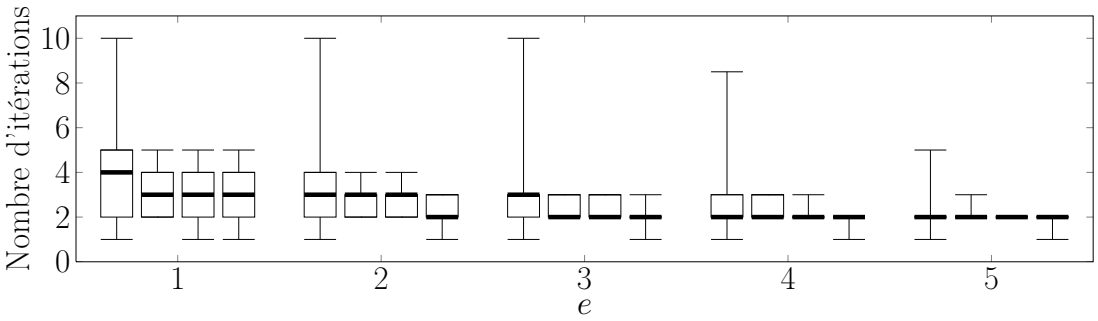



FIGURE *3.17 – Nombre d'itérations pour la convergence de la méthode ABiGlassoMaxLoop pour différentes valeurs de e et différentes valeurs de n. Pour une valeur de e, de gauche à droite, les boîtes à moustaches correspondent à* $n = 7$, $n = 12$, $n = 60$ *et* $n = 600$.

| | $e$ | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| $n = 60$ | 8 | 5 | 4 | 4 | 3 |
| $n = 600$ | 9 | 5 | 4 | 3 | 3 |

TABLEAU *3.3 – Nombre d'itérations maximal pour la convergence de ABiGlassoMaxLoop en fonction de e et de n.*

même pour $e$. La figure 3.17 montre le nombre d'itérations sur les 800 processus pour différentes valeurs de $n$ et de $e$.

Nous remarquons que pour la majorité des processus, la méthode converge en 2 itérations. Cela signifie que la solution retenue n'est pas $\widehat{G}_\lambda^{max}$ mais appartient à son voisinage de paramètre $e$.

Le nombre maximal d'itérations est limité à 10 ce qui explique que le nombre d'itérations ne soit pas supérieur à 10. Les configurations pour lesquelles il aurait été nécessaire d'avoir plus de 10 itérations sont pour $n = 7$, cas où le nombre d'observations est trop faible pour avoir un résultat pertinent.

Le tableau 3.3 donne le nombre d'itérations maximal en fonction de $e$ et $n$. Le nombre d'observations $n$ n'influe pas sur cette valeur contrairement à $e$. Plus $e$ est élevé, plus le nombre maximal d'itérations pour atteindre la convergence est faible. En effet, en une itération, plus de graphes sont comparés et donc il y a plus de probabilités de trouver le graphe le plus probable en un nombre réduit d'itérations.

Sachant que le temps de comparaison d'un voisinage de paramètre $e$ pour un processus à $p$ variables est en $o(p^{(e+2)})$, pour qu'il soit plus intéressant de parcourir successivement des voisinages de paramètre $e + 1$, il faut que le rapport entre le nombre d'itérations pour $e$ et pour $e+1$ soit supérieur à $p$. Sur les $800 \times 4$ processus que nous étudions, ce rapport vaut au maximum 4.5. Nous en concluons que, en terme de temps de calcul, il est plus intéressant de prendre $e$ le plus petit possible.

Le nombre d'itérations pour la convergence de la méthode ABiGlassoMaxLoop dépend aussi de $step_\Lambda$ car quand ce paramètre augmente, moins de graphes $\widehat{G}_\lambda$ sont générés, il y a donc moins de chance que le graphe qui maximise le score $z$ soit proche de la structure de graphe attendue, il faut alors faire plus d'itérations lors de l'étape de comparaison. $step_\Lambda$ choisi suffisamment petit permet de limiter le nombre d'itérations.

#### 3.3.3.3 Discussion compromis temps de calcul et qualité de la solution obtenue

La méthode donnant les meilleures performances en terme de distance de Hamming entre la solution et la structure de graphe attendue est la variante ABiGlassoMaxLoop. De plus, les performances sont les mêmes quelle que soit la valeur de $e$. Ce dernier point est important car plus $e$ est grand, plus il y a de graphes à comparer et donc plus la méthode est coûteuse en temps de calcul.

Les variantes ont été introduites pour réduire le temps de calcul de la méthode ABiGlasso de base. Le temps de calcul de la méthode ABiGlasso avec réduction d'intervalle est difficile à estimer et peut s'avérer plus important que celui de la méthode de base si le nombre d'itérations pour la recherche de l'intervalle réduit est trop important. Pour $e$ fixé, la méthode ABiGlassoMax est plus rapide que la méthode de base, l'étape de comparaison pouvant être jusqu'à $|\Lambda|$ fois plus rapide. La méthode ABiGlassoMaxLoop est également plus rapide que l'étape de base si le nombre d'itérations pour atteindre la convergence n'est pas trop élevé.

La méthode ABiGlasso de base, même pour $e = 5$, n'atteint pas les performances de la variante ABiGlassoMaxLoop en terme de distance à la structure attendue. D'un point de vue temps de calcul, puisque pour la variante ABiGlassoMaxLoop le nombre d'itérations pour atteindre la convergence est limité à 10, il est évident que cette variante avec $e = 1$ est plus rapide que la méthode de base avec $e = 5$. La variante ABiGlassoMax a des performances en terme de distance à la structure attendue similaires à celles d'ABiGlassoMaxLoop si $e = 5$. Pour que la variante ABiGlassoMax avec $e = 5$ soit plus intéressante en terme de temps de calcul que la variante ABiGlassoMaxLoop avec $e = 1$, il faut que la méthode ABiGlassoMaxLoop nécessite plus de $p^4$ itérations pour converger. Sachant que le nombre d'itérations pour atteindre la convergence est limité à 10, la méthode ABiGlassoMaxLoop est la méthode la plus performante à la fois pour la qualité de l'estimation de la structure d'indépendance conditionnelle et pour le temps de calcul.

Concernant la valeur de $step_\Lambda$, les performances de ABiGlassoMaxLoop ne sont pas impactées par cette valeur grâce à son aspect itératif. Nous avons choisi d'utiliser 0.1 car cette valeur permet de balayer un grand nombre de paramètres de pénalisation tout en gardant une complexité abordable. Si $step_\Lambda$ est trop petit, le temps de calcul est trop long à cause de $t_{Gl} \times |\Lambda|$ et si $step_\Lambda$ est trop grand il faut faire plus d'itérations pour que ABiGlassoMaxLoop converge et cela peut s'avérer problématique pour $p$ grand.

## 3.4 Discussion

Nous avons introduit une nouvelle méthode, la méthode ABiGlasso (Asymptotic Bayesian method initialized by Graphical lasso). Cette méthode combine de façon intelligente la méthode du Graphical lasso et un score qui permet d'évaluer parmi un ensemble de graphes lequel est le plus représentatif de la structure d'indépendance conditionnelle du processus étudié. Plusieurs variantes de cette méthode ont été présentées et comparées, l'objectif étant d'obtenir une méthode donnant une estimée la plus proche possible en terme de distance de Hamming de la structure d'indépendance conditionnelle pour un temps de calcul raisonnable. Nous avons mis en avant la variante ABiGlassoMaxLoop avec pour paramètres d'entrée $step_\Lambda = 0.1$ et $e = 1$ car c'est cette variante qui donne le meilleur compromis entre le temps de calcul et la performance d'estimation.

Nous avons également conclu que les méthodes ABiGlasso ne donnent pas de résultats pertinents quand $n \sim p$. Les résultats que nous avons jugés pertinents sont ceux pour lesquels $n \geq 10 \times p$ tout en ayant un regard critique sur les solutions obtenues pour $n \sim 10 \times p$, pour $p = 6$.

Dans ce chapitre, les méthodes ont été éprouvées sur des processus pour lesquels $p = 6$. Il est possible d'utiliser des valeurs de $p$ plus importantes. Il faut noter qu'augmenter $p$ n'augmente pas que le nombre de voisins à comparer mais aussi le temps de calcul de $V(G)$. En utilisant ABiGlassoMaxLoop avec $step_\Lambda = 0.1$, $e = 1$, il faut quelques secondes pour $p = 6$, quelques minutes pour $p = 10$ et une vingtaine de minutes pour $p = 20$.

# Chapitre 4

# Études comparatives de méthodes d'estimation de modèles graphiques gaussiens

**Sommaire**

Dans le chapitre précédent, nous avons introduit ABiGlasso, une nouvelle méthode d'estimation de modèles graphiques gaussiens, avec plusieurs variantes. Nous avons présenté et comparé les performances de ces différentes variantes. Dans ce chapitre nous continuons l'étude de la méthode ABiGlasso en comparant ses performances avec des méthodes existantes présentées dans le chapitre 1. L'objectif de cette comparaison est de positionner notre méthode par rapport aux méthodes existantes sur plusieurs critères : le temps de calcul, la qualité de la structure d'indépendance conditionnelle estimée, la capacité de la structure estimée à améliorer l'estimation des matrices de corrélation et des corrélations partielles et la capacité à évaluer la pertinence de la solution proposée (en utilisant un score soit sur la structure dans son ensemble soit sur les arêtes). Nous nous attardons sur les résultats obtenus pour les méthodes dont l'ensemble des solutions est limité à l'ensemble des graphes décomposables.

## 4.1 Modalités de comparaison

### 4.1.1 Méthodes comparées

Nous comparons les performances des méthodes suivantes :

1. Méthode de choix aléatoire du graphe solution (avec une probabilité de 0.5 sur chaque arête)
2. Méthode par seuillage de la matrice des corrélations partielles
3. Méthode Sure/Incertaine/Nulle (SIN)
4. Méthode Graphical lasso avec validation croisée (Glasso + CV)
5. Méthode *Bayesian Adaptive Graphical lasso* (BAGlasso)
6. Méthode de sélection de modèles graphiques gaussiens (GGMselect)
7. Méthode *Feature-Inclusion Stochastic Search* (FINCS)
8. Méthode *Asymptotic Bayesian method initialized by Graphical lasso* (ABiGlasso)

Ces méthodes ont été choisies car elles constituent un échantillon représentatif des méthodes rencontrées dans la littérature : la méthode aléatoire sert de référence pour juger les résultats obtenus pour les différentes méthodes, le seuillage de la matrice des corrélations partielles est une méthode basique, la méthode SIN [DP08] est une méthode d'estimation par tests-multiples sur la matrice des corrélations partielles, le Graphical lasso [FHT08] avec validation croisée est la méthode la plus couramment utilisée, elle fournit la matrice de précision parcimonieuse maximisant la log-vraisemblance pénalisée, les zéros de la matrice obtenue correspondant aux arêtes absentes dans le graphe recherché. La méthode BAGlasso [Wan12] est une variante du Graphical lasso pour laquelle le paramètre de pénalisation est recherché par une approche bayésienne. La méthode GGMselect [GHV12] est une méthode basée sur l'estimation des coefficients de régression entre les variables du processus, les coefficients nuls correspondant aux arêtes absentes du graphe recherché. La méthode FINCS [SC08] est une méthode d'estimation bayésienne sur l'ensemble des graphes décomposables et ABiGlasso est notre méthode qui recherche le graphe le plus probable sur un ensemble de graphes déterminé à l'aide du Graphical lasso.

Nous étudions également la méthode développée par Donnet et Marin [DM12] sur quelques exemples uniquement, pour des raisons d'initialisation : pour certains processus, nous n'avons pas réussi à franchir l'étape d'initialisation. Ayant la méthode FINCS de fonctionnelle, nous n'avons

effectué que quelques investigations pour identifier la raison de ce problème. Ces investigations n'ont pas permis de résoudre le problème rencontré.

Information Toolbox : Nous rappelons que plusieurs méthodes sont implémentées dans notre Toolbox. Ce sont les méthodes de seuillage *par_corr_thresh*, la méthode SIN *Method_sinUG_R*, la méthode du Graphical lasso *cov_sel*, la méthode BAGlasso *Method_BAGlasso*, la méthode GGMselect *Method_GGMselect_R* et la méthode ABiGlasso (nous utilisons ici la méthode ABiGlassoMaxLoop *Method_ABiGlasso_MaxLoop*).

Nous présentons maintenant les paramètres avec lesquels nous avons utilisé les méthodes pour mener l'étude comparative.

**Seuillage de la matrice des corrélations partielles** Comme les processus ont été générés avec $s = 0.2$, nous choisissons de seuiller les matrices des corrélations partielles avec ce même seuil.

**Méthode SIN** Les paramètres à régler sont les valeurs du risque de faux positif pour définir $G_S$ et $G_{SI}$ : nous considérons comme significatives les arêtes vérifiant un risque de faux positif inférieur à 0.01 et comme incertaines les arêtes vérifiant un risque de faux positif entre 0.01 et 0.9. Ces valeurs sont celles utilisées sur l'exemple donné dans l'article [DP08]. Dans la suite, nous nous intéressons uniquement au graphe $G_S$ car il donne de très bonnes performances.

**Graphical lasso avec validation croisée** Nous faisons une validation croisée 5-fold sur le nombre d'observations et nous parcourons l'ensemble des paramètres de pénalisation $\lambda = \{0.1, 0.2, \ldots, 10\}$. Nous conservons la solution qui maximise la log-vraisemblance. La valeur $\lambda = 0$ n'est pas parcourue car dans ce cas le Graphical lasso donne la matrice de précision empirique qui est l'estimateur maximum de vraisemblance et donc aucune des matrices générées pour un $\lambda \neq 0$ ne peut donner une log-vraisemblance plus élevée que celle calculée avec la matrice de précision empirique. La matrice de précision empirique n'est pas adaptée pour l'estimation de la structure d'indépendance conditionnelle car elle donne un graphe complet alors que nous sommes à la recherche d'une solution parcimonieuse.

**Bayesian Adaptive Graphical lasso** Pour cette méthode, nous utilisons les paramètres utilisés dans le code de démonstration fourni par Wang : 1000 itérations de chauffe et 2000 itérations MCMC. Nous utilisons également le seuillage proposé dans l'article de la méthode pour avoir une matrice des corrélations partielles parcimonieuse (cf chapitre 1, section 1.2.2.2). Ce seuillage est utilisé avec les paramètres donnés dans l'article : seuil de 0.5 et 1000 itérations pour générer les corrélations partielles moyennes.

**Méthode GGMselect** Nous utilisons cette méthode avec la famille LA (pénalisation lasso). Nous choisissons cette famille car elle correspond à une approche similaire à la nôtre c'est-à-dire que l'ensemble des graphes à comparer est un ensemble de solutions obtenues par pénalisation $\ell_1$, ici sur les coefficients de régression.

**FINCS** Nous utilisons les paramètres utilisés dans le code de démonstration fourni en complément de l'article [SC08], c'est-à-dire 3 000 000 itérations, un mouvement global toutes les 1000 itérations, un ré-échantillonnage d'un ancien modèle toutes les 20 itérations, nous conservons les 1000 graphes les plus probables, $g = 1/n$, l'a priori $\mathcal{HIW}$ a 1 degré de liberté et $\tau I$ comme matrice d'échelle avec $\tau = 0.02$, les arêtes ont une probabilité de 0.5 et les informations sont affichées toutes les 100 itérations.

**ABiGlasso** Nous utilisons la méthode ABiGlassoMaxLoop avec $step_\Lambda = 0.1$ et $e = 1$. Nous limitons le nombre d'itérations de convergence à 10.

### 4.1.2 Aspects comparés

Dans ce chapitre nous comparons les méthodes selon différents critères. Les critères retenus sont présentés ci-dessous avec la justification du choix de chaque critère.

**Temps de calcul** C'est l'un des aspects essentiels des méthodes d'estimation de modèles graphiques gaussiens. En effet, nous avons vu dans le chapitre 1 que le temps de calcul peut être rédhibitoire pour certaines méthodes quand le nombre de variables du processus étudié est important. Ceci est vrai pour les méthodes bayésiennes.

**Estimée de la structure d'indépendance conditionnelle** L'objectif de toutes les méthodes est d'estimer la structure d'indépendance conditionnelle des processus étudiés. Il est donc nécessaire de quantifier la distance entre la structure estimée et la structure attendue. Pour cela, nous utilisons la distance de Hamming, la sensibilité et la spécificité qui sont les métriques présentées dans le chapitre 2.

**Amélioration de l'estimation des matrices de corrélation et des corrélations partielles** Les méthodes Graphical lasso et BAGlasso donnent une matrice à partir de laquelle est construite la structure d'indépendance conditionnelle estimée. L'objectif de ces méthodes est de fournir une matrice donnant une meilleure estimation des matrices de corrélation et des corrélations partielles. Il existe un algorithme permettant de donner une meilleure estimée de ces matrices en utilisant la structure d'indépendance conditionnelle estimée, nous pouvons donc ainsi comparer l'ensemble des méthodes sur ce critère. Plus de détails sont donnés dans la section 4.4.

**Évaluation de la pertinence de la solution proposée** Plusieurs des méthodes existantes ainsi que la méthode ABiGlasso utilisent un score ou un critère permettant de dire que la solution proposée est plus pertinente que d'autres graphes envisagés comme solution. Ces méthodes sont la méthode GGMselect et les méthodes bayésiennes. Dans la section 4.5, nous étudions comment sont utilisés ces scores et quelle est l'information qu'ils apportent. D'autres méthodes donnent des scores sur les arêtes. C'est le cas de la méthode SIN qui donne la $p-$valeur de chacune des valeurs de corrélation partielle à l'issue du test multiple de l'hypothèse selon laquelle la valeur considérée est nulle. Scott *et al.* [SC08] ont également développé un score sur les arêtes. Toujours dans la section 4.5, nous étudions ces méthodes proposant un score sur les arêtes et nous introduisons également un score sur les arêtes pour la méthode ABiGlasso que nous comparons aux scores des autres méthodes.

En plus de différents critères, nous nous intéressons au comportement des méthodes pour lesquelles l'ensemble des solutions est limité à l'ensemble des graphes décomposables. Ces méthodes sont la méthode FINCS et celle de Donnet et Marin. Pour ces deux méthodes, nous faisons une étude approfondie des critères d'estimation de la structure d'indépendance conditionnelle et d'amélioration de l'estimation des matrices de corrélation et des corrélations partielles.

## 4.2 Comparaison du temps de calcul

Il est difficile de comparer les performances en termes de temps de calcul des différentes méthodes car elles ont été implémentées dans des langages différents. Les méthodes de seuillage de la matrice des corrélations partielles, le BAGlasso, le Graphical lasso avec validation croisée, la méthode de Donnet et Marin et ABiGlasso sont implémentées en Matlab®, la méthode FINCS est implémentée en C, la méthode SIN est implémentée en R et la méthode GGMselect est implémentée en R avec une partie des codes en C pour améliorer les performances en terme de temps de calcul.

Cependant, voici un ordre de grandeur des temps de calcul dans le cas $p = 6$ : les méthodes de seuillage, SIN et GGMselect sont très rapides, de l'ordre de la seconde. La méthode BAGlasso requiert quelques secondes pour fournir un résultat, tout comme la méthode ABiGlassoMaxLoop. La méthode du Graphical lasso avec validation croisée peut prendre plusieurs secondes si l'ensemble des paramètres de pénalisation est important (entre 10 secondes et 1 minutes pour 100 valeurs de $\lambda$, en fonction du processus). La méthode FINCS prend de deux jusqu'à une dizaine de minutes par processus.

## 4.3 Comparaison de la structure d'indépendance conditionnelle estimée

Nous comparons les performances des méthodes en terme de distance entre le graphe solution de chaque méthode et la structure attendue c'est-à-dire la structure imposée au processus simulé étudié. Pour quantifier cette distance, nous utilisons la distance de Hamming. Nous complétons l'étude en observant si les méthodes estiment mieux les arêtes présentes ou les arêtes absentes en utilisant les mesures de sensibilité et de spécificité. Nous regarderons aussi plus en détails les méthodes dont l'ensemble des solutions est limité aux graphes décomposables.

Information Toolbox : Nous rappelons que la fonction *comp_ graphs* de notre Toolbox permet de calculer la sensibilité, la spécificité et la distance de Hamming entre deux graphes.

### 4.3.1 Comparaison sur la distance de Hamming

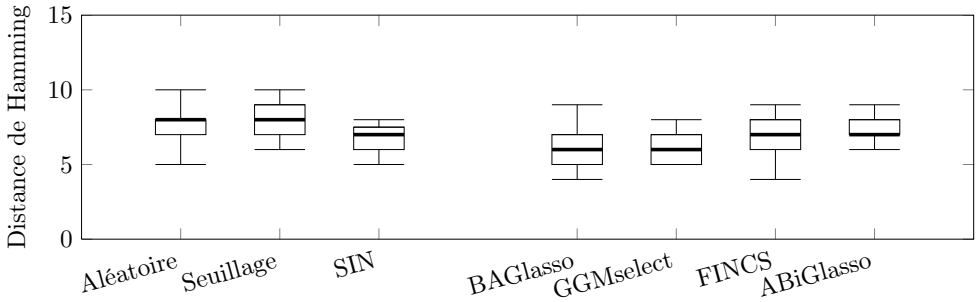
La figure 4.1 donne les résultats en terme de distance de Hamming entre la structure estimée et la structure attendue, sur les processus simulés présentés dans le chapitre 3. Chaque vignette correspond à un nombre d'observations $n$ donné. Les boîtes à moustaches sont construites de la façon suivante : le rectangle va du premier quartile (25% des données) au troisième quartile (75% des données), il est coupé par la médiane (en gras) et les moustaches vont du premier décile (10% des données) au neuvième décile (90% des données).

Pour $n = 7$, quelle que soit la méthode, les performances d'estimation de la structure d'indépendance conditionnelle sont similaires à un choix aléatoire du graphe solution. Nous n'avons pas appliqué la méthode Graphical lasso avec validation croisée pour $n = 7$ car ce nombre d'observations est trop faible pour faire une validation croisée 5-fold.

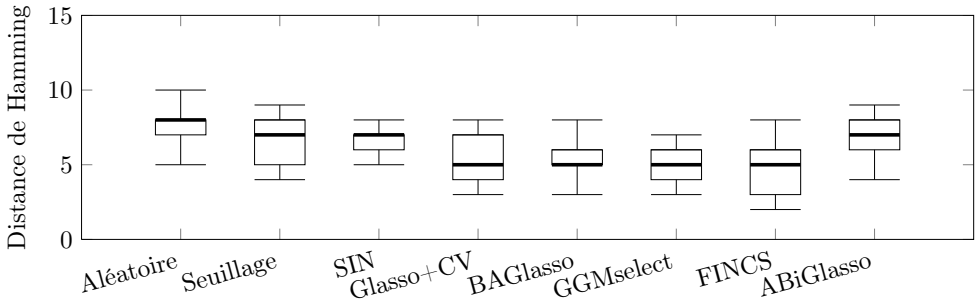
Pour $n = 12$, les performances sont légèrement meilleures que si le graphe solution avait été choisi aléatoirement : la médiane de la distance de Hamming entre le graphe solution et la structure attendue peut descendre à 5 pour les méthodes Graphical lasso avec validation croisée, BAGlasso, GGMselect et FINCS contre 8 pour le choix aléatoire. Cependant les graphes solutions sont encore loin de la structure attendue.

Pour $n = 60$, les méthodes ont une distance de Hamming médiane de 2 sauf les méthodes SIN, Graphical lasso avec validation croisée et BAGlasso qui ont une médiane plus élevée, valant jusqu'à 5 pour le Graphical lasso avec validation croisée. A l'exception de ces trois méthodes, les résultats obtenus sont corrects : pour plus d'un quart des données, la structure d'indépendance conditionnelle est correctement estimée et pour plus des trois quarts de données, la distance de Hamming entre la structure attendue et le graphe solution est inférieure ou égale à 3. Dans cette configuration, il faut porter un regard critique sur le graphe obtenu : il contient beaucoup d'information sur la structure d'indépendance conditionnelle mais il ne lui est probablement pas identique.

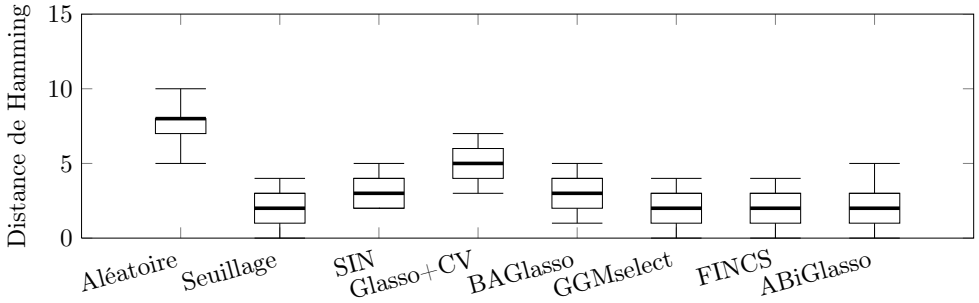
Pour $n = 600$, les méthodes de seuillage, SIN, GGMselect et ABiGlasso ont leur médiane à 0. La méthode SIN donne même le graphe attendu pour plus de 90% des données, et les méthodes GGMselect et ABiGlasso ont une distance de Hamming inférieure ou égale à 1 pour plus de 90% des processus. La méthode FINCS donne des résultats un peu moins bons (un quart des données ont pour solution un graphe qui diffère d'au moins 2 arêtes avec la structure attendue). Cela vient du fait que la méthode FINCS donne comme solution un graphe décomposable, nous étudions ce point plus en détail par la suite (cf section 4.3.3). Notons également que la méthode du Graphical lasso avec validation croisée donne très rarement la structure attendue (plus de 90% des données ont une distance de Hamming supérieure ou égale à 1) et que le BAGlasso a pour solution un graphe qui diffère d'au moins une arête dans plus de la moitié des cas. Cela



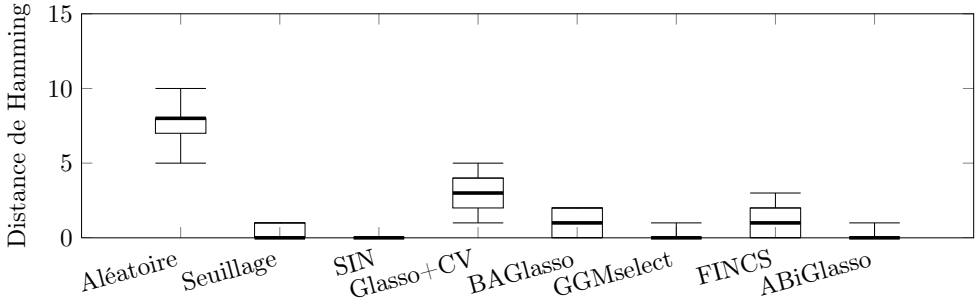

(a) $n = 7$




(b) $n = 12$




(c) $n = 60$




(d) $n = 600$

FIGURE *4.1 – Boîtes à moustaches sur la distance de Hamming pour les* 800 *processus simulés pour différentes valeurs du nombre d'observations n et pour les 8 méthodes énumérées section 4.1.1.*

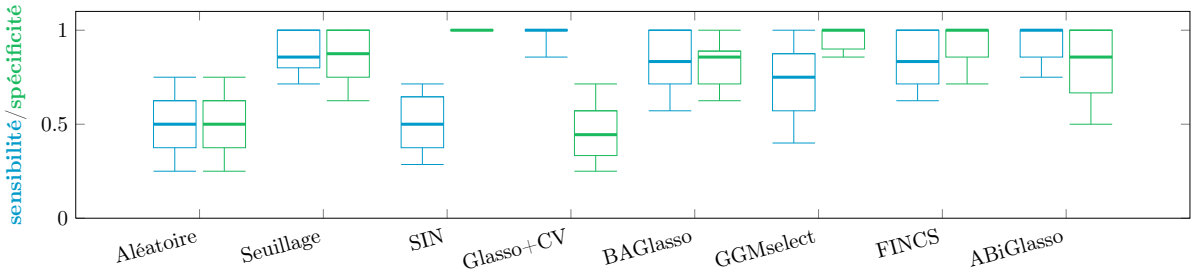
illustre la non convergence du Graphical lasso. Pour les autres méthodes, la solution représente la structure d'indépendance conditionnelle ou en diffère de très peu d'arêtes.

Pour compléter cette étude, nous cherchons à savoir si les arêtes les mieux estimées sont les

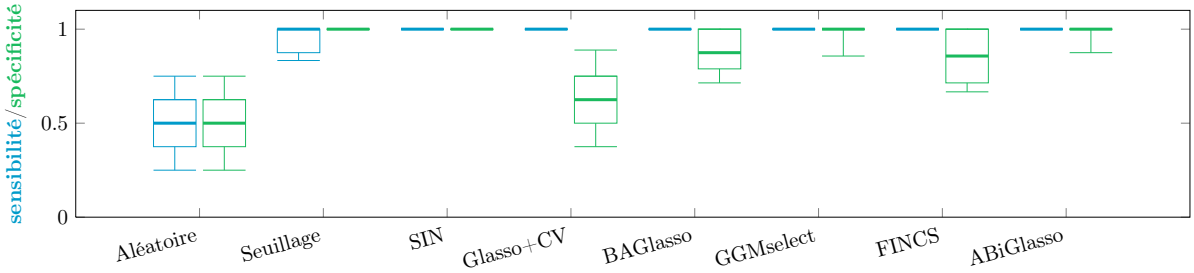
arêtes présentes ou les arêtes absentes. Pour cela nous étudions les performances en terme de sensibilité et de spécificité dans la section suivante.

### 4.3.2 Comparaison sur la sensibilité et la spécificité

Comme les performances pour $n = 7$ et $n = 12$ sont proches de l'aléatoire, nous réalisons l'étude de la sensibilité et la spécificité uniquement pour $n = 60$ et $n = 600$. La figure 4.2 montre les valeurs de ces deux mesures sur les 800 processus simulés pour les différentes méthodes et les $n$ retenus.




(a) $n = 60$




(b) $n = 600$

FIGURE *4.2 – Sensibilité (en bleu) et spécificité (en vert) entre les solutions des 800 processus et les structures attendues pour les 8 méthodes que nous comparons, pour* $n = 60$ *et* $n = 600$.

Nous rappelons que plus la sensibilité est proche de 1, meilleure est l'estimation des arêtes attendues et plus la spécificité est proche de 1, meilleure est l'estimation des arêtes censées être absentes.

Pour $n = 60$, les méthodes estimant mieux les arêtes absentes que les arêtes présentes sont les méthodes SIN, GGMselect et BAGlasso. Les solutions de ces méthodes sont donc en général trop parcimonieuses, elles contiennent moins d'arêtes que la structure attendue. Les méthodes estimant mieux les arêtes présentes que les arêtes absentes sont les méthodes Graphical lasso avec validation croisée et ABiGlasso. Les solutions de ces deux méthodes ne sont donc en général pas assez parcimonieuses, elles contiennent plus d'arêtes que la structure attendue. Les méthodes de seuillage et BAGlasso ont une sensibilité et une spécificité similaires, ce sont indifféremment des arêtes présentes ou non attendues qui sont mal estimées. Cette étude permet de savoir comment critiquer la solution obtenue pour $n = 60$ : pour les méthodes qui en général donnent une solution avec trop d'arêtes, nous savons que l'ensemble des arêtes de la structure attendue est contenu dans l'ensemble des arêtes de la solution proposée et pour les méthodes qui en général proposent une solution trop parcimonieuse, nous savons que l'ensemble des arêtes de la solution proposée est contenu dans l'ensemble des arêtes de la structure de graphe attendue. Notons que pour la méthode SIN, la sensibilité est similaire à celle obtenue dans le cas d'un choix aléatoire du graphe solution et que pour la méthode Graphical lasso avec validation croisée, la spécificité est même

légèrement moins bonne que dans le cas d'un choix aléatoire du graphe solution.

Pour $n = 600$, à l'exception de la méthode de seuillage, les arêtes attendues sont bien estimées. Cela signifie que quelle que soit la méthode considérée, le graphe solution contient la structure attendue. Notons que pour la méthode Graphical lasso avec validation croisée, la spécificité tend vers la spécificité obtenue pour un choix aléatoire du graphe solution.

Les moins bonnes performances de la méthode SIN pour $n = 60$ s'expliquent ainsi : pour $n = 60$, la valeur à partir de laquelle une corrélation partielle est significativement différente de zéro est plus élevée que pour $n = 600$, or pour construire le graphe contenant uniquement les arêtes significatives (c'est-à-dire les corrélations partielles significativement différentes de zéro) nous gardons le même risque de faux positif pour $n = 60$ et $n = 600$. $G_S$ contient donc moins d'arêtes que la structure attendue quand $n = 60$. Les moins bonnes performances ne dépendent pas que du choix du risque de faux positif. En effet, la méthode se base sur les valeurs de la matrice des corrélations partielles empirique. A $n = 60$ l'estimation de cette matrice est de moins bonne qualité que pour $n = 600$ et cela impacte directement les performances de la méthode.

### 4.3.3 Méthodes à solutions dans l'espace des graphes décomposables

La méthode FINCS est une méthode dont l'ensemble des solutions est limité à l'ensemble des graphes décomposables. Parmi les structures imposées à nos processus simulés, 4 sont non-décomposables, les structures 1, 5, 6 et 7. La figure 4.3 donne les performances de la méthode FINCS pour chacune des 8 structures imposées (pour $n = 600$). Pour les 4 structures décomposables, la méthode FINCS a pour solution la structure imposée. Pour les structures non décomposables, la distance de Hamming médiane varie selon la structure. D'après la figure 4.2, la méthode FINCS a une spécificité non égale à 1 pour certains processus, cela signifie que cette méthode donne une solution ayant plus d'arêtes que la structure attendue. Pour chaque structure, la distance de Hamming médiane correspond au nombre d'arêtes qu'il faut ajouter pour rendre la structure décomposable. Par contre les arêtes ajoutées ne sont pas les mêmes d'un processus à l'autre, même pour une même structure.

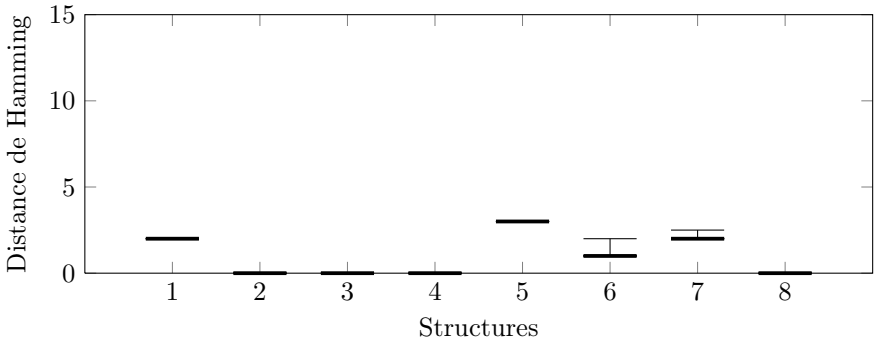



FIGURE *4.3 – Boîtes à moustaches sur la distance de Hamming pour la méthode FINCS. Pour chaque structure, la méthode est appliquée sur 100 processus avec* $n = 600$. *La méthode donne d'excellents résultats pour les processus dont la structure attendue est décomposable (2, 3, 4 et 8) et moins bons pour les autres.*

Une autre méthode ayant pour ensemble de solutions l'ensemble des graphes décomposables est la méthode de Donnet et Marin [DM12]. Nous avons appliqué cette méthode uniquement aux processus générés à partir des structures 1, 2, 4 et 5 et avec $n = 600$ à cause de problèmes d'initialisation. Les résultats sur ces 4 structures sont présentés dans la figure 4.4. Les observations sont les mêmes que pour la méthode FINCS.

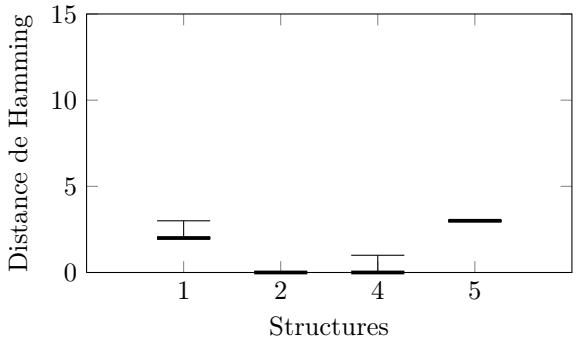



FIGURE *4.4 – Boîtes à moustaches sur la distance de Hamming pour la méthode de Donnet et Marin. Pour les structures 1, 2, 4 et 5, la méthode est appliquée sur 100 processus avec* $n = 600$*. La méthode donne d'excellents résultats pour les processus dont la structure attendue est décomposable (2 et 4) et moins bons pour les autres.*

### 4.3.4 Conclusion

Les méthodes comparées ont toutes des performances de faible qualité pour $n \sim p$. Pour $n = 60$, les solutions obtenues sont correctes mais diffèrent de quelques arêtes avec la structure attendue. En fonction des méthodes, trois cas de figure se présentent, permettant en général de critiquer la solution obtenue :

- pour certaines méthodes seules les arêtes présentes peuvent être considérées comme appartenant à la structure d'indépendance conditionnelle
- pour d'autres seules les arêtes absentes peuvent être considérées comme n'appartenant pas à la structure d'indépendance conditionnelle
- pour les méthodes restantes il n'est pas possible de savoir quel type d'arêtes est fiable et quel type d'arête ne l'est pas.

Pour $n = 600$, à l'exception du Graphical lasso avec validation croisée et de la méthode FINCS, toutes les méthodes donnent comme solution la structure d'indépendance conditionnelle ou une structure très proche en terme de distance de Hamming. La méthode ABiGlasso a des performances similaires aux meilleures méthodes existantes.

## 4.4 Comparaison sur l'amélioration de l'estimation des matrices de corrélation et des corrélations partielles

Notre méthode ABiGlasso a été développée pour extraire la structure d'indépendance conditionnelle d'un processus. Mais parmi les méthodes utilisées, comme le BAGlasso ou le Graphical lasso, l'objectif est de fournir une meilleure estimation de la matrice de précision (et donc des matrices de corrélation et des corrélations partielles).

### 4.4.1 Amélioration de l'estimation des matrices de corrélation et des corrélations partielles

Il est possible, à partir d'un graphe, d'améliorer l'estimation des matrices de corrélation et des corrélations partielles au sens du maximum de vraisemblance.

L'estimation de la matrice de covariance au sens du maximum de vraisemblance s'obtient en imposant que si deux variables sont reliées par une arête leur covariance est égale à leur covariance empirique et que si elles ne le sont pas que leur corrélation partielle est nulle. Si le graphe est décomposable, l'estimateur du maximum de vraisemblance est obtenu explicitement

et est donc facile à calculer. Si le graphe n'est pas décomposable, le problème est plus compliqué. Il existe cependant des algorithmes pour calculer cette estimée (cf Algorithme 17.1 dans [HTF09] auquel nous ferons référence sous l'appellation *algorithme de HTF*).

Information Toolbox : L'algorithme de HTF est présent dans notre Toolbox. C'est la fonction *MLEg_fct* qui calcule une matrice de covariance dont l'inverse respecte la structure d'indépendance conditionnelle et la matrice de covariance empirique données en entrée de la fonction.

Nous choisissons donc d'étudier les performances des méthodes sur les matrices de corrélation $C$ et des corrélations partielles $\Pi$ qui sont respectivement les versions normalisées de matrices de covariance $\Sigma$ et de précision $K$. Nous utilisons l'algorithme de HTF pour fournir une estimée de la matrice de covariance quand la méthode donne directement un graphe.

Si la structure utilisée pour améliorer l'estimation a des arêtes qui ne devraient pas être présentes ou si il lui manque des arêtes, les estimées des matrices de covariance et de précision peuvent être plus éloignées des matrices théoriques que les matrices empiriques. Fitch et Jones [FJ12] ont mis en lumière ce phénomène en évaluant l'impact sur l'estimation des matrices de covariance et de précision de l'approximation d'un graphe non décomposable par un graphe décomposable. Leur étude se limite à des cas où des arêtes sont ajoutées pour rendre un graphe décomposable. Pour la matrice de précision, la variance est légèrement impactée par cet ajout d'arêtes. Pour la matrice de covariance, l'ajout d'arêtes peut se traduire par une matrice de covariance estimée très éloignée de la matrice de covariance attendue.

### 4.4.2 Métriques utilisées pour l'évaluation de l'influence de l'utilisation de la structure d'indépendance conditionnelle estimée

Pour évaluer la distance entre deux matrices, nous utilisons la norme $\ell_1$ : $||\cdot||_1$. Le choix de cette norme est arbitraire. Pour évaluer si l'estimée de la matrice prenant en compte le graphe solution de la méthode considérée est meilleure que l'estimée empirique, nous divisons la distance entre la matrice attendue et la matrice estimée à partir d'un graphe par la distance entre la matrice attendue et la matrice empirique. La matrice attendue, c'est-à-dire celle utilisée pour simuler les processus, est notée $C$ pour la matrice de corrélation et $\Pi$ pour la matrice des corrélations partielles, l'estimée empirique est notée $\widehat{C}_{emp}$ et l'estimée à partir du graphe $\widehat{C}_G$, de même pour $\Pi$. Nous travaillons à partir des rapports :

$$R_C = \frac{||C - \widehat{C}_G||_1}{||C - \widehat{C}_{emp}||_1} \text{ et } R_\Pi = \frac{||\Pi - \widehat{\Pi}_G||_1}{||\Pi - \widehat{\Pi}_{emp}||_1}$$

### 4.4.3 Évaluation sur les processus simulés

L'utilisation du graphe est censée améliorer l'estimation de la matrice donc le rapport doit être inférieur à 1. La figure 4.5 et la figure 4.6 donnent respectivement les valeurs des rapports $R_\Pi$ et $R_C$ pour chacune des 8 méthodes comparées sur les 800 processus simulés avec $n = 600$.

Considérons dans un premier temps la figure 4.5. Concernant les méthodes pour lesquelles la matrice des corrélations partielles est obtenue en utilisant l'algorithme de HTF avec le graphe solution, les rapports $R_\Pi$ supérieurs à 1 correspondent à des processus pour lesquels le graphe solution diffère d'au moins une arête avec la structure attendue, tous les graphes correctement estimés améliorent l'estimation de $\Pi$.

Pour les méthode estimant directement $\Pi$, c'est-à-dire le Graphical lasso et le BAGlasso, les performances sont moins bonnes : plus de la moitié des processus ont une estimée $\widehat{\Pi}_G$ moins bonne que $\widehat{\Pi}_{emp}$ même quand le graphe solution est identique à la structure attendue. Pour ces

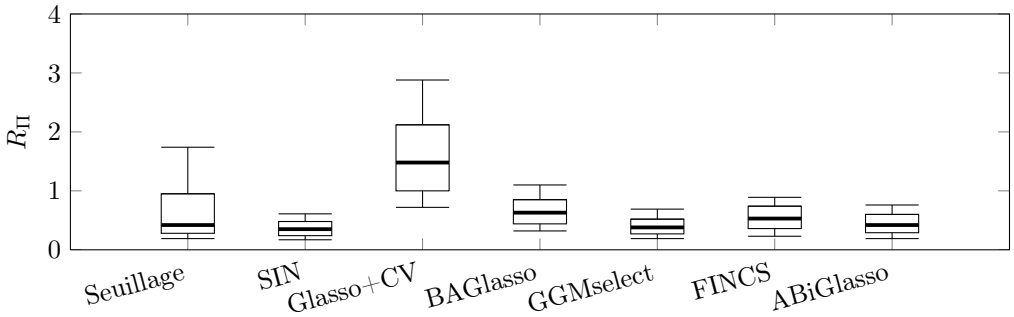



FIGURE *4.5 – Rapport $R_\Pi$ entre la matrice des corrélations partielles obtenue et celle attendue, pour $n = 600$.*

deux méthodes, si la matrice des corrélations partielles est calculée en utilisant l'algorithme de HTF avec le graphe solution alors les résultats sont meilleurs : si le graphe est correctement estimé, le rapport $R_\Pi$ est inférieur à 1. Cependant si le graphe n'est pas correctement estimé ou si $R_\Pi$ valait déjà moins de 1 quand le graphe était bien estimé alors les performances sont en moyenne meilleures mais pas obligatoirement.

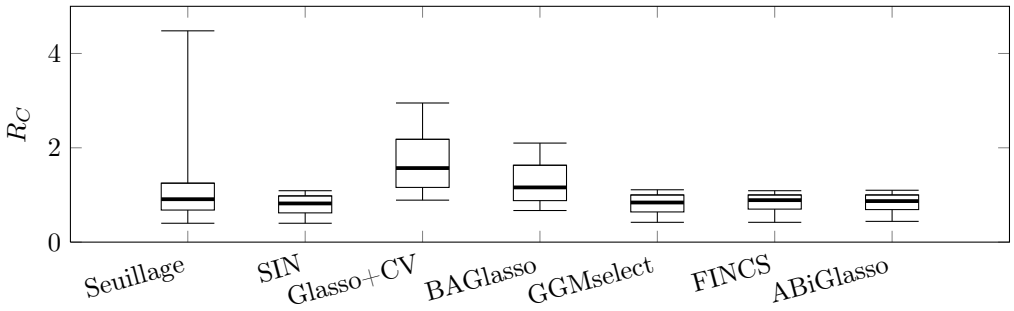



FIGURE *4.6 – Rapport $R_C$ entre la matrice de corrélation obtenue et celle attendue, pour $n = 600$.*

Considérons maintenant la figure 4.6. Quelle que soit la méthode considérée, les rapports $R_C$ sont moins bons que les rapport $R_\Pi$. Comme pour $\Pi$, les méthodes Graphical lasso avec validation croisée et BAGlasso donnent les moins bons résultats. La méthode de seuillage donne de bons résultats concernant l'estimation de la structure mais ne donne pas de très bons résultats pour l'estimation de la matrice des corrélations, le rapport maximal pouvant atteindre 60.6. Nous avons constaté que ces rapports élevés correspondent à des matrices de corrélations empiriques ayant un déterminant très faible.

En complément de la figure 4.6, pour la méthode Graphical lasso avec validation croisée, certains rapports peuvent atteindre des valeurs de l'ordre de $10^3$. Ces rapports sont observés quand la solution du Graphical lasso a une distance de Hamming très importante avec la structure attendue et que le déterminant de la matrice des corrélations empirique est proche de 0.

### 4.4.4 Méthodes à solutions dans l'espace des graphes décomposables

Pour la méthode FINCS, l'estimation de la structure est limitée aux structures décomposables. Nous étudions comment cette limitation impacte l'estimation de la matrice de corrélation. La figure 4.7 montre la valeur du rapport $R_C$ sur les structures décomposables et sur les structures non décomposables.

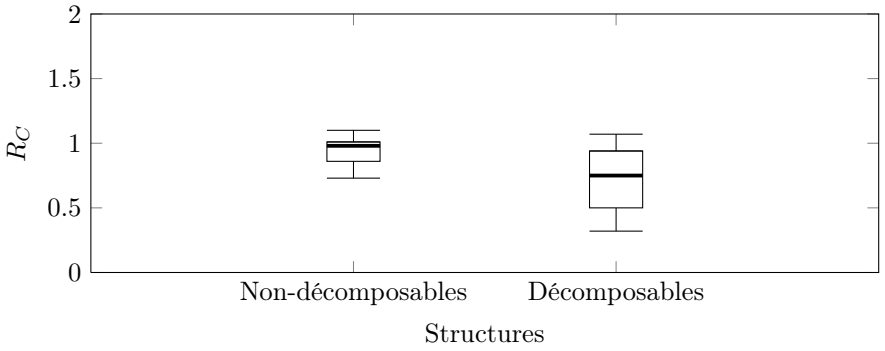



FIGURE *4.7 – Rapport $R_C$ entre la matrice de corrélation obtenue et celle attendue pour la méthode FINCS et $n = 600$.*

Les performances sont meilleures quand la structure attendue est décomposables car $R_C$ est plus proche de 0 mais dans les deux cas plus de 75% des données ont un rapport $R_C$ inférieur ou égal à 1.

### 4.4.5 Conclusion

L'étude des performances sur l'amélioration de l'estimation des matrices de corrélation et des corrélations partielles n'apporte pas d'information complémentaire à l'estimation de la structure d'indépendance conditionnelle : l'estimée dépend du graphe solution. Si la structure est bien estimée, la matrice est plus proche de la matrice attendue que la matrice empirique et si la structure est mal estimée cela peut induire une matrice très éloignée de la matrice attendue.

## 4.5 Comparaison sur l'évaluation de la pertinence de la solution proposée

La méthode ABiGlasso a l'avantage de proposer un score $z(G|\mathbf{X})$ proportionnel à la probabilité du graphe $G$ de représenter la structure d'indépendance conditionnelle du processus considéré $\mathbf{X}$. Ce score peut être utilisé de plusieurs façons : pour comparer différents graphes ou pour juger de la pertinence d'une arête.

La méthode ABiGlasso n'est pas la seule à proposer un score sur les graphes, les méthodes bayésiennes ainsi que la méthode GGMselect le font également. De plus, les méthodes SIN et FINCS proposent un score sur les arêtes.

### 4.5.1 Scores sur les graphes

La méthode GGMselect [GHV12] sélectionne le graphe le plus probable en minimisant un critère (cf chapitre 1, section 1.2.3). Cependant seul le critère pour le graphe solution est donné à l'utilisateur et non les critères des autres graphes de la famille considérée. La méthode FINCS [SC08] fournit également un score sur les graphes les plus probables gardés en mémoire : le logarithme de la probabilité a posteriori de chaque graphe. Cependant cette méthode se limite aux graphes décomposables. Les autres méthodes ne fournissent pas de scores permettant de comparer les graphes.

Le score $z$, introduit avec la méthode ABiGlasso (cf equation 3.2), permet de comparer les graphes entre eux et donc de savoir si le graphe solution est nettement plus probable que les autres ou si plusieurs structures ont des scores très proches.

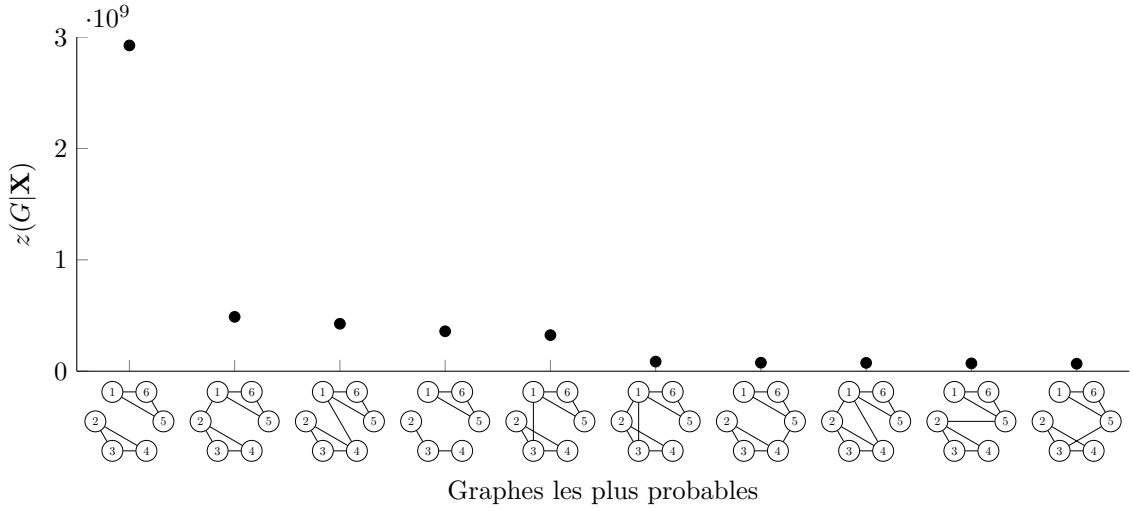



FIGURE *4.8 – Scores des 10 graphes ayant les plus forts scores. Le graphe le plus probable correspond à la structure d'indépendance conditionnelle attendue. Pour ce processus, le graphe ayant le plus fort score se distingue aisément des autres.*

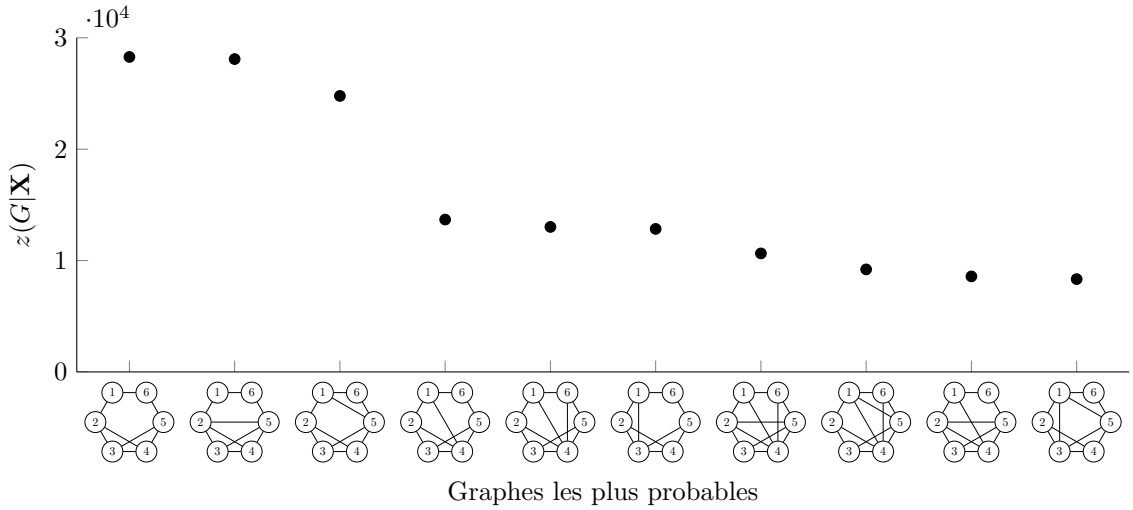



FIGURE *4.9 – Scores des 10 graphes ayant les plus forts scores. Le graphe le plus probable correspond à la structure d'indépendance conditionnelle attendue. Pour ce processus, les second et troisième graphes ont un score proche de celui du graphe solution.*

Les figures 4.8 et 4.9 montrent les scores des 10 graphes ayant les scores les plus élevés pour deux processus distincts avec $n = 600$. Pour le processus de la première figure, le graphe proposé a un score plus fort que les graphes suivants. Ce graphe est la représentation de la structure d'indépendance conditionnelle. Pour le deuxième processus, les scores du deuxième et du troisième graphes sont proches de celui du graphe proposé comme solution. Les trois graphes peuvent donc être considérés comme des représentants de la structure d'indépendance conditionnelle du processus étudié, cela signifie que les arêtes qui diffèrent entre les trois graphes n'apportent pas d'information sur la structure d'indépendance conditionnelle du processus.

### 4.5.2 Scores sur les arêtes

#### 4.5.2.1 Score sur les arêtes à partir du score utilisé dans la méthode ABiGlasso

Les scores $z(G|\mathbf{X})$ peuvent aussi être utilisés pour juger de la pertinence d'une arête.

**Définition 4.5.1.** ***Score d'une arête :*** *pour chaque arête $(i,j)$ reliant les nœuds $i$ et $j$, nous définissons le score d'une arête de la façon suivante :*

$$z\big((i,j)|\mathbf{X},\mathcal{G}\big) = \frac{\sum_{G\in\mathcal{G}} z(G|\mathbf{X}) \times \mathbb{1}_{(i,j)\in E}}{\sum_{G\in\mathcal{G}} z(G|\mathbf{X})}.$$

*où $\mathcal{G}$ est un sous-ensemble de graphes.*

Le score d'une arête dépend de l'ensemble de graphes $\mathcal{G}$ considéré : le score $z((i,j)|\mathbf{X},\mathcal{G})$ est le ratio entre les scores des graphes contenant l'arête $(i,j)$ et appartenant à $\mathcal{G}$ et les scores de tous les graphes de $\mathcal{G}$.

Le score $z\big((i,j)|\mathbf{X},\mathcal{G}\big)$ est compris entre 0 et 1. Si il est proche de 1, cela signifie que l'arête est soit présente dans beaucoup des graphes parcourus, soit dans les graphes les plus probables, soit les deux. Si il est proche de 0, cela signifie que l'arête est soit dans très peu des graphes parcourus, soit dans les graphes les moins probables, soit les deux.

#### 4.5.2.2 Performances en utilisant notre score sur les arêtes

Les figures 4.10 et 4.11 présentent les scores des différentes arêtes pour les 8 structures utilisées pour simuler nos données. Pour chaque structure, les boîtes à moustaches sont tracées à partir de 100 processus. Nous comparons les résultats obtenus pour $n = 60$ (en orange) et $n = 600$ (en gris).

Pour $n = 600$, le score des arêtes attendues est très proche de 1, la plus petite valeur rencontrée valant 0.97 toutes structures et toutes arêtes confondues. Pour les arêtes non attendues, dans la majorité des cas le score est proche de 0 mais pour certains processus ce score peut être proche de 1 : parmi les 800 processus, 51 arêtes ont un score supérieur à 0.9. Nous observons que toutes les arêtes ayant un score supérieur à 0.62 appartiennent à la solution retenue par ABiGlasso, qui par conséquent diffère d'au moins une arête avec la structure attendue. Nous observons également que toute arête ayant un score inférieur à 0.359 est absente des solutions ABiGlasso.

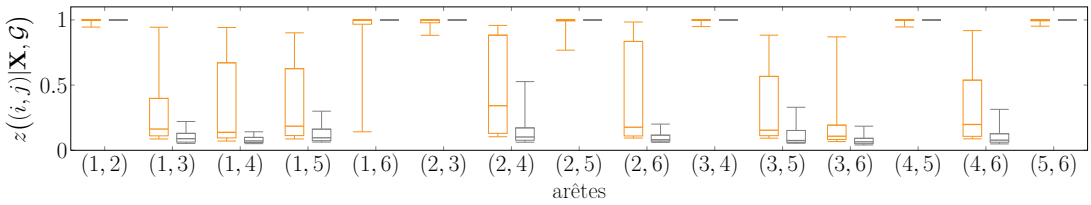
Pour $n = 60$, les scores des arêtes présentes sont dans la majorité des cas proches de 1 (pour plus de 75% des données). Par contre, les scores des arêtes absentes sont beaucoup plus hétérogènes même si la médiane des scores d'une arête absente vaut au plus 0.35.

Le fait que l'utilisation du score sur les arêtes montre que les arêtes présentes sont mieux estimées que les arêtes absentes concorde avec les observations faites à partir de la sensibilité et la spécificité (figure 4.2) : les solutions ABiGlasso ont une sensibilité meilleure que leur spécificité ce qui signifie que les arêtes attendues sont bien estimées mais que les arêtes non attendues le sont moins.

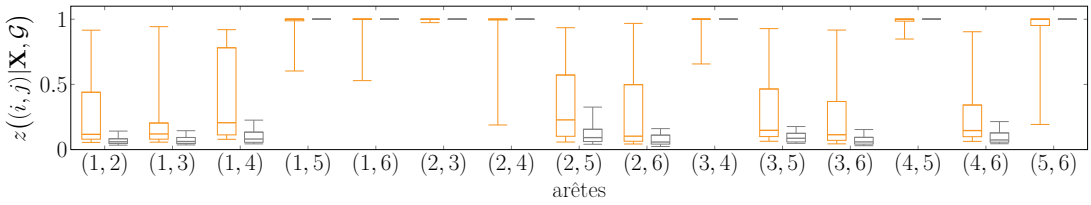
#### 4.5.2.3 Comparaison des méthodes proposant un score sur les arêtes

ABiGlasso n'est pas la seule méthode à fournir de l'information sur la pertinence des arêtes. La méthode FINCS [SC08] fournit une probabilité d'inclusion $\widehat{\mathbb{P}}_{ij}$ pour chaque arête $(i,j)$ :

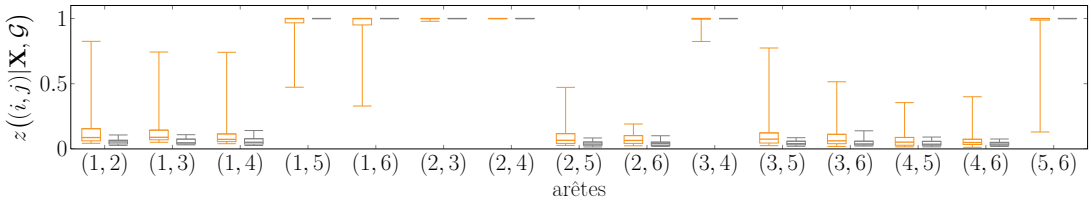
$$\widehat{\mathbb{P}}_{ij} = \frac{\sum_{G\in\mathcal{G}_{best}} \mathbb{P}(\mathbf{X}|G)\pi(G) \times \mathbb{1}_{(i,j)\in E}}{\sum_{G\in\mathcal{G}_{best}} \mathbb{P}(\mathbf{X}|G)\pi(G)}. \tag{4.1}$$



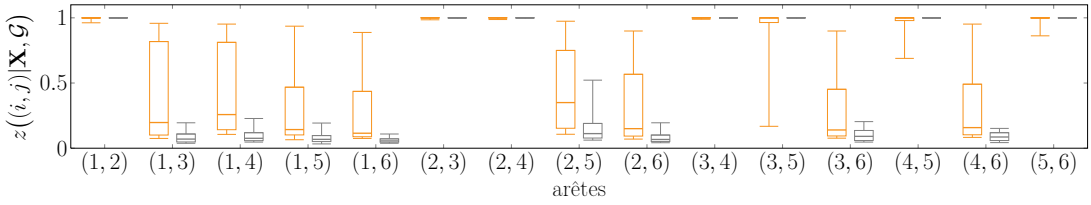

(a) Structure 1




(b) Structure 2




(c) Structure 3



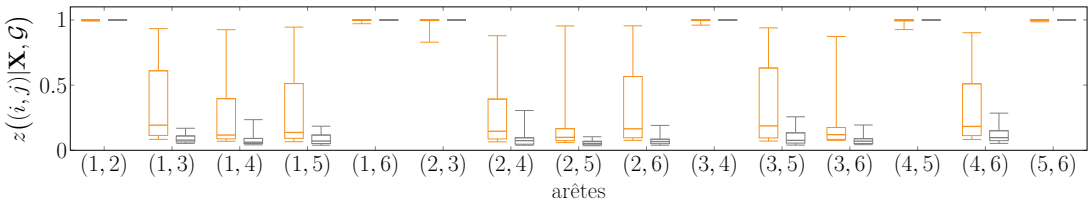

(d) Structure 4

FIGURE *4.10 – Scores des arêtes pour les structures imposées 1 à 4 avec* $n = 60$ *(en orange) et* $n = 600$ *(en gris). Les arêtes attendues ont leur médiane proche de 1 et les arêtes non attendues proche de 0 quel que soit n ou la structure considérée. Par contre, pour* $n = 60$*, le score des arêtes non attendues peut être proche de 1.*

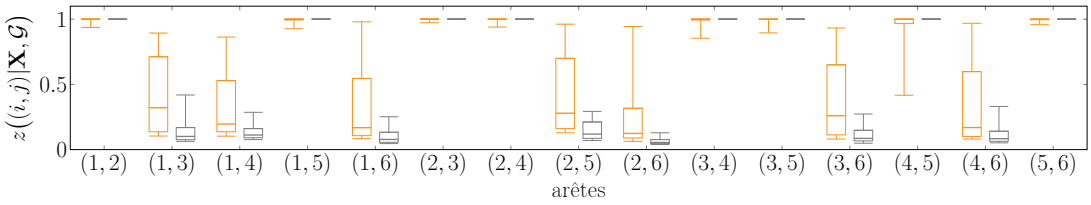
où $\mathbb{P}(\mathbf{X}|G)$ est la probabilité d'avoir le processus étudié $\mathbf{X}$ sachant le graphe $G$, $\pi(G)$ est l'a priori sur $G$ et $\mathcal{G}_{best}$ est l'ensemble des graphes les plus probables conservés lors de l'application de la méthode FINCS à $\mathbf{X}$. Bien entendu, les graphes dans $\mathcal{G}_{best}$ sont uniquement des graphes décomposables. La probabilité d'inclusion est construite sur le même principe que le score sur les arêtes que nous avons proposé à partir du score $z$. Quoique antérieure à nos travaux, elle ne nous a pas influencé pour l'établissement de notre score sur les arêtes mais nous a conforté sur la façon dont nous avons construit notre propre score.

La méthode SIN [DP08] est basée sur une approche par tests multiples sur les valeurs de corrélations partielles : si la $p$-valeur associée à une valeur de corrélation partielle est très faible, la corrélation partielle est significativement non nulle et donc l'arête associée est significative. Par contre si la $p$-valeur est élevée alors la corrélation partielle est très certainement nulle et l'arête associée est significativement absente.

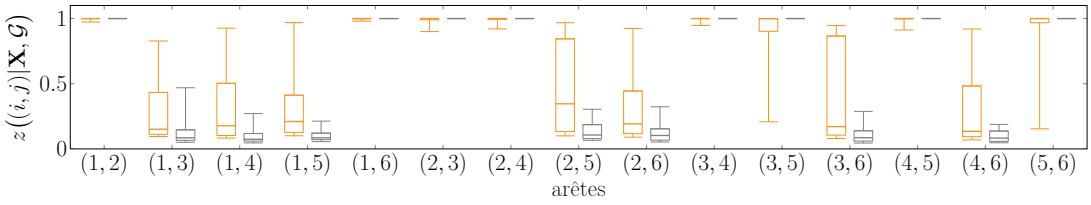
Comme pour la méthode SIN, nous pouvons utiliser le score sur les arêtes pour la méthode



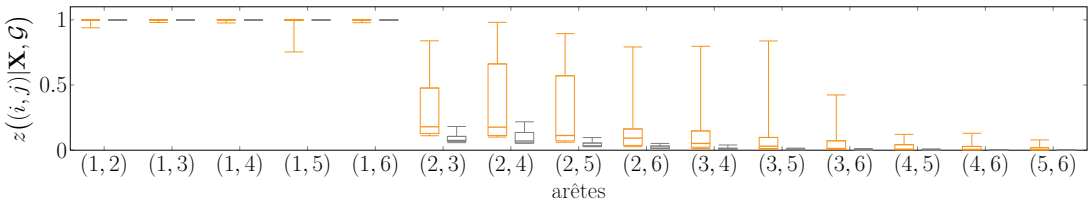

(a) Structure 5




(b) Structure 6




(c) Structure 7




(d) Structure 8

FIGURE *4.11 – Scores des arêtes pour les structures imposées 5 à 8 avec* $n = 60$ *(en orange) et* $n = 600$ *(en gris). Les arêtes attendues ont leur médiane proche de 1 et les arêtes non attendues proche de 0 quel que soit* $n$ *ou la structure considérée. Par contre, pour* $n = 60$*, le score des arêtes non attendues peut être proche de 1.*

ABiGlasso et la probabilité d'inclusion pour la méthode FINCS pour construire deux graphes : un contenant les arêtes sûres $G_S$ et un contenant les arêtes sûres et incertaines $G_{SI}$. Ces deux graphes se construisent en imposant deux seuils sur les données, par exemple en considérant comme sûres les arêtes ayant un score (ou une probabilité d'inclusion) supérieur à 0.99 et comme incertaines les arêtes ayant un score (ou une probabilité d'inclusion) entre 0.1 et 0.99. Nous comparons les résultats obtenus en utilisant cette approche aux résultats obtenus avec la méthode SIN sur les 800 processus pour $n = 60$ et $n = 600$ (cf tableau 4.1). Quel que soit $n$, les performances de ABiGlasso avec le score sur arêtes sont meilleures que celles de SIN et celles de FINCS avec la probabilité d'inclusion. Pour $n = 60$ la méthode FINCS donne de meilleurs résultats que la méthode SIN mais pour $n = 600$ c'est l'inverse. Cela signifie que en se concentrant sur l'estimation arête par arêtes plutôt que sur l'estimation de la structure dans son ensemble, la méthode ABiGlasso est la plus performante.

La figure 4.12 compare l'approche ABiGlasso avec score sur arêtes avec la méthode ABiGlasso simple. Nous observons que l'utilisation du score sur arêtes améliore les performances de la méthode ABiGlasso, quelle que soit la valeur de $n$.

| | $n = 60$ | | | $n = 600$ | | |
|---|---|---|---|---|---|---|
| | ABiGlasso | FINCS | SIN | ABiGlasso | FINCS | SIN |
| $G_S = G$ | 182 | 27 | 11 | 794 | 781 | 790 |
| $E(G_S) \subset E(G) \subset E(G_{SI})$ | 393 | 620 | 391 | 2 | 8 | 4 |
| autres | 97 | 147 | 395 | 4 | 11 | 6 |

TABLEAU *4.1 – Comparaison des résultats obtenus pour les méthodes SIN, FINCS avec probabilité d'inclusion et ABiGlasso avec score sur arêtes pour les 800 processus et* $n = 60$ *et* $n = 600$*. La première ligne donne le nombre de graphes pour lesquels le graphe* $G_S$ *correspond à la structure d'indépendance conditionnelle attendue* $G$ *pour les différentes configurations. La deuxième ligne donne le nombre de graphes pour lesquels les arêtes du graphe* $G_S$ *sont contenues dans la structure attendue et les arêtes de la structure attendue sont contenues dans* $G_{SI}$ *mais* $G \neq G_S$ *et* $G \neq G_{SI}$*. La troisième ligne donne le nombre de graphes ne vérifiant ni l'égalité de la première ligne, ni l'encadrement de la deuxième ligne.*

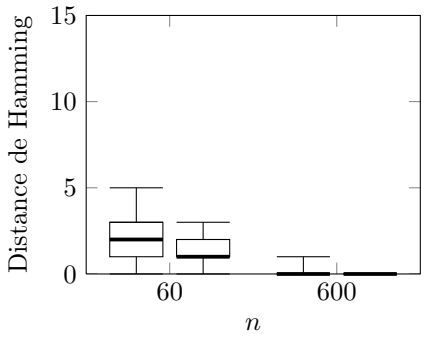



FIGURE *4.12 – Distance de Hamming pour la méthode ABiGlasso simple et la méthode ABiGlasso avec score sur les arêtes, pour* $n = 60$ *et* $n = 600$*, sur les 800 processus simulés. Pour chaque valeur de* $n$*, la boîte à moustache de gauche correspond à la méthode ABiGlasso simple et celle de droite à la méthode ABiGlasso avec score sur les arêtes.*

### 4.5.3 Conclusion

Le score $z$ permet à la méthode ABiGlasso d'être compétitive face aux méthodes proposant un score sur la structure estimée dans son ensemble ou un score sur les arêtes permettant de définir une structure contenant les arêtes définies assurément présentes dans la structure et une autre contenant toutes les arêtes sauf celles définies assurément non présentes. Grâce à ces scores, la méthode ABiGlasso est un outil pertinent pour comparer différentes structures de graphes.

## 4.6 Comparaison sur un exemple à 100 variables

Toutes les études comparatives menées dans ce chapitre (ainsi que les autres études menées dans les différents chapitres de cette thèse) sont réalisées sur des processus à $p = 6$ variables. Dans cette section, nous illustrons sur un exemple comment se comportent les méthodes pour un nombre de variables beaucoup plus important : $p = 100$.

### 4.6.1 Processus

Nous choisissons de travailler avec différentes valeurs de $n$ :
- $n = 100p$ pour le cas $n >> p$
- $n = 10p$ pour le cas $n > p$
- $n = 2p$ pour le cas $n \sim p$

Pour $p = 100$ et $n = 100p$, une valeur de corrélation partielle est statistiquement significativement non nulle si elle est supérieure à 0.033 en valeur absolue (pour un risque de faux positif $\alpha = 0.001$). Pour $p = 100$ et $n = 10p$, une valeur de corrélation partielle est statistiquement significativement non nulle si elle est supérieure à 0.109 en valeur absolue (toujours pour un risque de faux positif $\alpha = 0.001$). Nous souhaitons générer une matrice des corrélations partielles respectant une structure d'indépendance conditionnelle donnée et dont les valeurs non nulles sont statistiquement significativement non nulles dans le cas $n = 100p$ mais pas nécessairement dans le cas $n = 10p$ (comme pour les processus à 6 variables que nous avons étudiés jusqu'à présent). Nous choisissons de générer une matrice des corrélations partielles dont les valeurs non nulles sont supérieures en valeur absolue à 0.05.

Nous n'avons pas réussi à générer une matrice des corrélations partielles qui respecte une structure d'indépendance conditionnelle donnée et dont les valeurs non nulles sont supérieures en valeur absolue au seuil $s = 0.05$ en utilisant notre approche (2.1.2, chapitre 2). Nous avons donc utilisé la méthode de Castelo présentée dans le chapitre 2. Dans un premier temps, nous avons essayé de générer des matrices jusqu'à en avoir une dont les valeurs non nulles sont toutes supérieures en valeur absolue au seuil $s = 0.05$ mais nous n'avons pas réussi à trouver une telle matrice. Nous avons donc pris une des matrices générées. La matrice obtenue respecte bien une structure d'indépendance conditionnelle donnée mais ses valeurs non nulles ne sont pas toujours statistiquement non nulles.

Puisque nous ne pouvons pas contrôler que les valeurs non nulles soient significativement différentes de zéro, nous choisissons de comparer les structures que nous obtenons dans les différentes configurations et avec les différentes méthodes avec la structure obtenue en seuillant la matrice des corrélations partielles générée par la méthode de Castelo avec un seuil $s = 0.05$. Toutes les corrélations partielles supérieures à $s$ en valeur absolue correspondent à une arête, les autres correspondent à une absence d'arête. Notons que nous ne savons pas si la structure obtenue par seuillage est la structure attendue mais elle en ait plus proche que celle imposée en entrée de la méthode de Castelo.

### 4.6.2 Méthodes

Nous utilisons toutes les méthodes comparées dans ce chapitre à l'exception de la méthode FINCS. La méthode FINCS est trop longue en temps de calcul : par exemple, sur le processus avec $n = 100p$, il faut environ 3h30 pour faire 5 000 itérations, nous évaluons donc qu'il faut environ trois mois pour effectuer le calcul sur 3 000 000 itérations.

### 4.6.3 Résultats

Nous étudions les résultats obtenus sur les trois processus ($p = 100$ et $n \in \{2p, 10p, 100p\}$).

#### 4.6.3.1 Temps de calcul

Le tableau 4.2 donne le temps de calcul pour les différentes méthodes.

| Méthode | Seuillage | SIN | Glasso+CV | BAGlasso | GGMselect | ABiGlasso |
|---|---|---|---|---|---|---|
| Temps | 0.02 sec | 1s | 28 min | 5 min | 20 min | 14 j |

TABLEAU *4.2 – Temps de calcul moyen sur les trois processus pour obtenir une estimée de la structure d'indépendance conditionnelle de ces processus*

Le temps de calcul des méthodes de seuillage et SIN n'est pas impacté par le nombre de variables, il reste de l'ordre de la seconde. Pour la méthode BAGlasso, le temps de calcul prend quelques minutes au lieu de quelques secondes mais cette méthode reste très rapide. Les méthodes

Glasso+CV et GGMselect ont un temps de calcul de l'ordre de la vingtaine de minutes contre quelques secondes pour $p = 6$. Si l'étude à réaliser comporte un nombre restreint de processus et n'exige pas un résultat très rapidement, ces deux méthodes restent accessibles en terme de temps de calcul. La méthode ABiGlassoMaxLoop avec comme paramètre de voisinage $e = 1$ requiert quatorze jours pour fournir un résultat sur un unique processus. Ce temps est trop long pour mener une étude sur différents processus.

Le passage à un nombre de variables plus élevé rend notre méthode inadaptée. L'augmentation du temps de calcul est dûe à l'augmentation du nombre de graphes à comparer (pour $e = 1$, il faut comparer $10 \times \left(1 + \frac{p(p-1)}{2}\right)$ graphes) et à l'augmentation du temps de calcul du score $z$ (pour $p = 6$, $t_z \simeq 0.12s$ et pour $p = 100$, $t_z \simeq 25s$). Le tableau 4.3 donne des valeurs de $t_z$ pour différentes valeurs de $p$. Pour $p = 20$, le temps de calcul reste abordable (environ une vingtaine de minutes pour avoir une solution), pour $p = 50$ il faut compter environ douze heures pour obtenir une solution et cela est trop coûteux en temps de calcul si on veut étudier plusieurs processus.

| $p$ | 6 | 20 | 50 | 100 |
|---|---|---|---|---|
| $t_z$ (en secondes) | 0.12 | 0.56 | 3.35 | 25.5 |

TABLEAU *4.3 – Exemple de temps de calcul du score z pour un graphe pour différentes valeurs de p.*

#### 4.6.3.2 Qualité de la solution obtenue

Pour évaluer la qualité de la solution obtenue, nous utilisons la distance de Hamming, la sensibilité et la spécificité entre l'estimée et la structure obtenue en seuillant la matrice des corrélations partielles générée par la méthode de Castelo. Comme nous ne savons pas si cette structure correspond exactement à la structure attendue, nous l'appelons structure de référence.

Pour la méthode ABiGlasso, nous avons uniquement les résultats pour le processus avec $n = 100p$ pour des raisons de temps de calcul.

Le tableau 4.4 donne la distance de Hamming entre les structures estimées et la structure de référence pour les différentes valeurs de $n$. La figure 4.13 complète le tableau en donnant la spécificité et la sensibilité pour les différentes méthodes et les différentes valeurs de $n$.

| Méthode | Aléatoire | Seuillage | SIN | Glasso+CV | BAGlasso | GGMselect | ABiGlasso |
|---|---|---|---|---|---|---|---|
| $n = 100p$ | 2475 | 1380 | 153 | 1231 | 1040 | 629 | 1596 |
| $n = 10p$ | 2475 | 1370 | 1211 | 1462 | 1140 | 728 | x |
| $n = 2p$ | 2475 | 1362 | 1461 | 1462 | 1807 | 1377 | x |

TABLEAU *4.4 – Distance de Hamming pour chacune des méthodes sur les processus donnés en fonction de n.*

Pour $n = 100p$, la méthode SIN donne le meilleur résultat, pour $n = 10p$, le meilleur résultat est donné par la méthode GGMselect.

Pour $n \sim p$ les résultats ne sont pas très bons même si ils restent meilleurs que les résultats obtenus aléatoirement.

Notons que quelle que soit la méthode, ce sont les arêtes absentes qui sont les mieux estimées comme en témoignent les valeurs élevées de spécificité dans la figure 4.13.

Le résultat obtenu pour ABiGlasso pour $n = 100p$ est mauvais, toutes les autres méthodes donnent un meilleur résultat. Cela s'explique par la construction de la méthode. Nous comparons

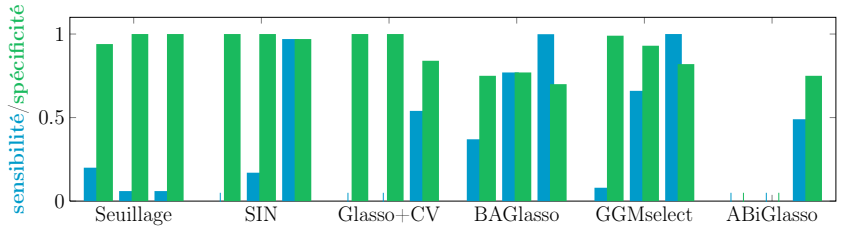



FIGURE *4.13 – Sensibilité et spécificité pour les différentes méthodes sur les différents processus. Pour une méthode, de gauche à droite, $n = 2p$, $n = 10p$ et $n = 100p$. Pour la méthode ABiGlasso nous avons uniquement les résultats pour $n = 100p$.*

uniquement des graphes qui diffèrent de très peu d'arêtes avec la solution du Graphical lasso la plus probable parmi les solutions données pour un ensemble de paramètres de pénalisation donné. Si la solution du Graphical lasso retenue n'est pas bonne, la méthode ABiGlasso ne pourra pas trouver la bonne structure si celle-ci est trop éloignée de la solution du Graphical lasso.

Il est important de garder à l'esprit que les métriques utilisées permettent d'évaluer approximativement les différentes méthodes puisque nous n'avons pas réussi à générer des processus dont nous connaissons exactement la structure d'indépendance conditionnelle.

### 4.6.4 Conclusion

Les résultats présentés dans cette partie sont obtenus à partir de trois processus donnés ayant différents nombres d'observations. Cette étude simple permet de mettre en lumière plusieurs aspects :

- nous n'avons pas réussi à générer des processus gaussiens multivariés dont nous connaissons la structure d'indépendance conditionnelle dans le cas $p = 100$
- la méthode FINCS a un temps de calcul rédhibitoire pour des cas où $p$ est grand (supérieur à quelques dizaines)
- la méthode ABiGlasso n'est pas adaptée aux cas où $p$ est grand (supérieur à quelques dizaines) aussi bien pour des raisons de temps de calcul que de performances d'estimation
- la méthode SIN est très rapide et donne de très bons résultats dans la configuration $n >> p$
- la méthode GGMselect donne des résultats corrects dans la configuration $n > p$ avec un temps de calcul d'une vingtaine de minutes

Ces différents aspects sont importants à prendre en compte si on souhaite faire des études sur des processus pour lesquels le nombre de variables est grand (de l'ordre de la centaine voire plus).

## 4.7 Discussion

Dans ce chapitre nous avons comparé différentes méthodes d'estimation de structures d'indépendance conditionnelle représentatives des méthodes existantes sur des processus à $p = 6$ variables. Concernant la qualité de la solution proposée, notre méthode ABiGlasso a des performances similaires à celles des meilleures méthodes quelle que soit la configuration dans laquelle nous nous plaçons : pour les configurations où la méthode ABiGlasso a de faibles performances, les autres méthodes n'ont pas de meilleures performances et dans les autres configurations elle est l'une des meilleures méthodes. Seule la méthode SIN donne de meilleurs résultats dans la configuration $p = 6$ et $n = 600$. Nous avons mis en évidence que l'amélioration de l'estimation des matrices de corrélation et des corrélations partielles est fortement liée à la qualité de l'es-

timation de la structure d'indépendance conditionnelle et donc que l'étude de l'estimée de ces matrices n'apportait pas d'informations supplémentaires à l'estimation des structures d'indépendance conditionnelle.

Nous avons également vu que la méthode ABiGlasso se démarque de la majorité des méthodes car elle possède un score qui permet de juger de la pertinence de la solution proposée, que ce soit de la structure dans son ensemble ou des arêtes indépendamment les unes des autres.

La méthode ABiGlasso n'est pas adaptée dans le cadre de données avec un nombre important de variables car cette méthode est très coûteuse en temps de calcul principalement à cause du nombre de graphes à comparer. Dans un cas idéal où $n >> p$, la méthode SIN donne de très bons résultats en une fraction de seconde. Par contre pour des configurations où $n > p$, la méthode GGMselect donne de meilleurs résultats en une vingtaine de minutes. Le choix de la méthode permettant d'obtenir une bonne estimée de la structure d'indépendance d'un processus est à faire en fonction de la configuration des processus étudiés. Notons que pour $n \sim p$, quelle que soit la méthode considérée, les résultats sont de mauvaise qualité.

# Chapitre 5

# Étude conjointe de processus à l'aide de leurs structures d'indépendance conditionnelle

**Sommaire**



Jusqu'à présent, nous nous sommes intéressés à l'estimation de la structure d'indépendance conditionnelle d'un processus. Nous avons vu qu'estimer cette structure est un problème complexe quand le nombre d'observations n'est pas très grand devant le nombre de variables, quelle que soit la méthode utilisée pour l'estimation. D'autre part, nous avons vu dans le chapitre précédent que la force de la méthode ABiGlasso repose sur le score $z$ (équation 5.1). Dans cette partie, nous souhaitons comparer des processus en utilisant leurs structures d'indépendance conditionnelle estimées. Pour cela, nous utilisons des profils de scores. Dans un premier temps nous explicitons les motivations d'une telle approche, ensuite, nous présentons comment construire des profils de scores utilisables pour faire de la comparaison de processus et enfin nous les utilisons pour classifier des groupes de processus synthétiques.

## 5.1 Motivation

Nous souhaitons comparer plusieurs processus à l'aide de leurs structures d'indépendance conditionnelle estimées. Plus exactement, nous souhaitons discriminer deux catégories de processus, comme par exemple les processus associés à des sujets sains et ceux associés à des sujets malades (cf chapitre 6). En utilisant directement les structures d'indépendance conditionnelle estimées, les outils à notre disposition sont les métriques sur graphes telles l'efficacité globale [LM01] ou le coefficient de clustering [WS98]. Ces métriques sont cependant pertinentes uniquement pour comparer des graphes ayant le même nombre d'arêtes [BB11]. Puisque aucun a priori n'a été fait sur les structures que nous estimons, les graphes solutions n'ont pas le même nombre d'arêtes d'un processus à l'autre. Il faut donc trouver une alternative aux métriques usuelles.

Dans le développement de la méthode ABiGlasso, nous avons introduit le score $z$ :

$$z(G|\mathbf{X}) = \frac{\varphi_{\boldsymbol{\pi}_{\bar{E}}, cW_{\bar{E}\bar{E}}}(\mathbf{0})}{V(G)}. \tag{5.1}$$

Ce score permet de quantifier, pour un processus donné $\mathbf{X}$, si un graphe est plus pertinent qu'un autre pour représenter la structure d'indépendance conditionnelle du processus.

Reprenons le processus exemple utilisé pour illustrer les méthodes ABiGlasso dans le chapitre 3. La figure 5.1 donne des exemples de graphes avec leurs scores associés pour ce processus. Le graphe ayant le score le plus élevé est identique à celui utilisé pour générer le processus. C'est ce phénomène qui est utilisé pour estimer la structure d'indépendance conditionnelle en utilisant ABiGlasso. Cependant les différents scores nous informent que le troisième graphe, même si il n'est pas celui avec le score le plus élevé, contient de l'information sur la structure d'indépendance conditionnelle du processus alors que le premier et le deuxième graphes n'apportent pas d'information sur cette structure.

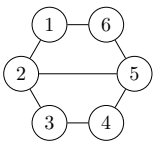
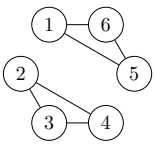
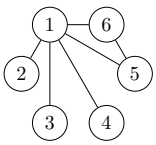
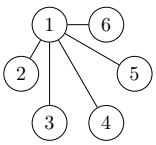
| graphe |  |  |  |  |
|---|---|---|---|---|
| score $z$ | 0 | 0 | $2.34 \times 10^9$ | $13.5 \times 10^9$ |



FIGURE *5.1 – Exemples de scores z sur quatre graphes pour un processus donné. Le graphe de droite a le plus fort score et correspond au graphe utilisé pour générer le processus étudié.*

En calculant le score $z$ pour la totalité des graphes possibles, nous pouvons tracer le profil de densité sur l'ensemble des graphes et ainsi voir quels graphes contiennent de l'information sur la structure d'indépendance conditionnelle du processus. Cependant, cela est trop coûteux en temps de calcul.

Nous choisissons de construire un profil de scores en calculant les scores sur un ensemble réduit de graphes afin de limiter le temps de calcul. Le choix du sous-ensemble de graphes à utiliser pour tracer un profil de scores est à faire en fonction de la façon dont le profil sera exploité par la suite.

## 5.2 Profils de scores

### 5.2.1 Étude d'un processus

Nous souhaitons étudier un seul processus et voir si la solution proposée, par exemple par la méthode ABiGlassoMaxLoop, est vraiment beaucoup plus probable que les autres ou si un groupe

de graphes est plus ou moins équiprobable, c'est-à-dire que tous les graphes appartenant à ce groupe sont des candidats pertinents pour représenter la structure d'indépendance conditionnelle du processus. Dans ce cas, le sous-ensemble de graphes que nous pouvons considérer pour tracer un profil de scores est le voisinage de la structure d'indépendance conditionnelle estimée. Dans le cas de l'utilisation de la méthode ABiGlassoMaxLoop pour estimer la structure d'indépendance conditionnelle, nous pouvons considérer l'ensemble des graphes parcourus lors des différentes itérations sur les voisinages successifs. C'est cette approche qui est utilisée dans le chapitre 4 section 4.5.1 pour tracer les profils des figures 4.8 et 4.9 même si uniquement les scores des 10 graphes les plus probables sont montrés dans les illustrations.

### 5.2.2 Comparaison de processus

Nous souhaitons comparer $k$ processus : $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(k)}$. Nous notons $\widehat{G}(\mathbf{X}^{(i)})$ l'estimée de la structure d'indépendance conditionnelle du processus $\mathbf{X}^{(i)}$. Dans l'ensemble de ce chapitre, nous utilisons l'équation (3.4) pour définir $\widehat{G}(\mathbf{X}^{(i)})$ mais d'autres définitions peuvent être utilisées. Si les graphes estimés ne sont pas les mêmes, les voisinages sont également différents. En utilisant l'approche décrite dans la section précédente pour construire le profil de scores associés à chaque processus, ces profils ne peuvent pas être comparés car ils ne sont pas construits sur le même ensemble de graphes. Il faut trouver un ensemble de graphes pertinent pour construire des profils de scores comparables.

#### 5.2.2.1 Profils croisés

Nous choisissons comme sous-ensemble de graphes pour le profil de scores l'ensemble des graphes estimés pour chacun des processus. En faisant ainsi, nous calculons $k$ profils (un par processus) sur $k$ graphes (un par processus).

**Définition 5.2.1.** ***Profils croisés :***
*Soient $k$ processus, les profils croisés correspondent aux profils de scores de chacun des $k$ processus sur l'ensemble de graphes formé des $k$ estimées de la structure d'indépendance conditionnelle, chacune associée à l'un des $k$ processus.*

Pour le processus $\mathbf{X}^{(i)}$, les valeurs de son profil croisé forment le vecteur

$$\left[z\left(\widehat{G}(\mathbf{X}^{(1)})|\mathbf{X}^{(i)}\right), z\left(\widehat{G}(\mathbf{X}^{(2)})|\mathbf{X}^{(i)}\right), \ldots, z\left(\widehat{G}(\mathbf{X}^{(k)})|\mathbf{X}^{(i)}\right)\right]. \tag{5.2}$$

Prenons par exemple le processus utilisé pour la figure 5.1. Supposons que nous souhaitons comparer $k = 10$ processus et que les structures d'indépendance conditionnelle estimées des processus soient données dans la figure 5.2.

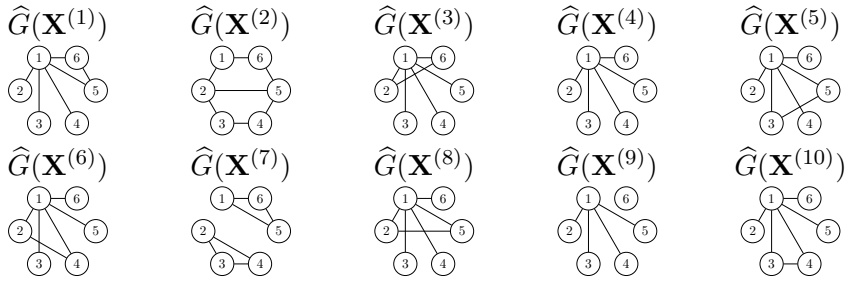



FIGURE 5.2 – *Structures d'indépendance conditionnelle estimées pour $k = 10$ processus. L'ensemble de ces graphes permet de construire le profil croisé de la figure 5.3*

Le processus que nous considérons est le processus $i = 4$. La figure 5.3 donne son profil croisé sur l'ensemble des graphes estimés des $k = 10$ processus que nous souhaitons comparer. La valeur la plus élevée du profil est obtenue pour $\widehat{G}(\mathbf{X}^{(4)})$ qui est bien l'estimée du processus considéré. Pour les graphes très éloignés en terme de distance de Hamming de $\widehat{G}(\mathbf{X}^{(4)})$, le score est nul. C'est le cas pour les graphes $\widehat{G}(\mathbf{X}^{(2)})$ et $\widehat{G}(\mathbf{X}^{(7)})$. Notons également que pour deux processus ayant la même distance de Hamming entre leurs structures estimées et la structure estimée du processus étudié, les valeurs prises par le score peuvent être très différentes. En effet, les graphes $\widehat{G}(\mathbf{X}^{(5)})$ et $\widehat{G}(\mathbf{X}^{(9)})$ ont une distance de Hamming de 1 avec le graphe $\widehat{G}(\mathbf{X}^{(4)})$ et le score du graphe $\widehat{G}(\mathbf{X}^{(5)})$ est très élevé alors que celui du graphe $\widehat{G}(\mathbf{X}^{(7)})$ est nul. Cela illustre le fait que la distance de Hamming entre la structure estimée d'un processus et la structure estimée du processus étudié n'est pas une métrique pertinente pour comparer des processus.

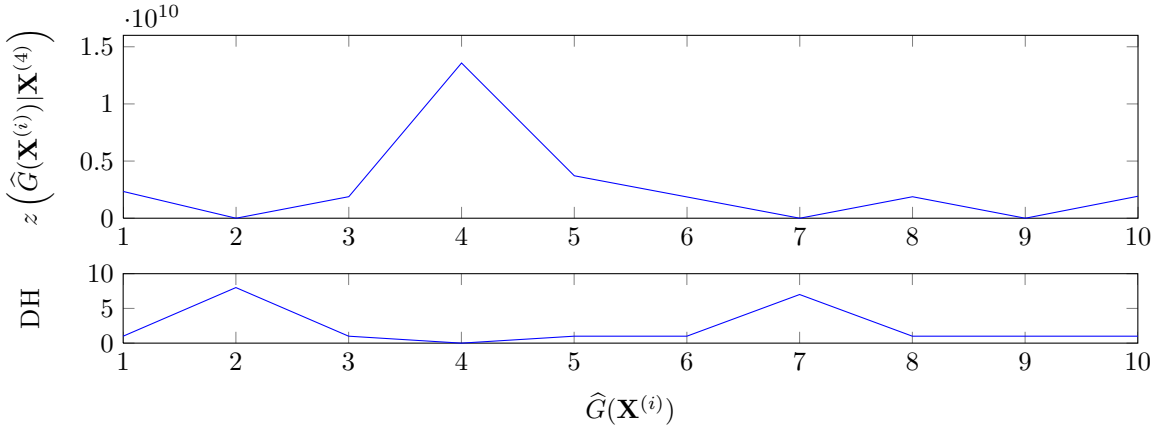



FIGURE *5.3 – Exemple de profil croisé pour le processus utilisé à la figure 5.1 sur la figure du haut. La figure du bas donne la distance de Hamming (DH) entre la structure estimée pour le processus étudié* $\mathbf{X}^{(4)}$ *et les structures estimées pour les* $k = 10$ *processus.*

Comme nous voulons comparer les processus, nous voulons également comparer les profils croisés. Or ces profils sont construits à partir des scores $z$ et les scores dépendent du processus utilisé pour les calculer. Il faut donc introduire une normalisation permettant la comparaison.

#### 5.2.2.2 Profils croisés normalisés

Nous choisissons de normaliser le profil de scores d'un processus de façon à avoir un profil de densité : nous normalisons par la somme des scores du profils.

**Définition 5.2.2.** ***Profils croisés normalisés :***
*Soient $k$ processus, les profils croisés normalisés correspondent aux profils croisés de ces $k$ processus, normalisés de façon à avoir des profils assimilables à des profils de densité (la somme des valeurs d'un profil vaut 1).*

Cette normalisation permet de comparer les tendances des différents profils. Un processus pour lequel un des $k$ graphes a un score très élevé alors que les autres ont des scores faibles va avoir pour profil de densité un pic proche de 1 et le reste proche de 0 alors qu'un processus pour lequel les $k$ graphes ont des scores similaires va avoir un profil croisé normalisé quasi-uniforme de valeur $1/k$.

Soit $\widehat{\mathcal{G}}$ l'ensemble des $k$ graphes associés aux $k$ processus, nous introduisons la fonction $Q_{\widehat{\mathcal{G}}}$ qui pour un processus $\mathbf{X}^{(i)}$ est définie ainsi :

$$Q_{\widehat{\mathcal{G}}}(\cdot|\mathbf{X}^{(i)}) = \frac{z\left(\cdot|\mathbf{X}^{(i)}\right)}{\sum_{G\in\widehat{\mathcal{G}}} z\left(G|\mathbf{X}^{(i)}\right)} \tag{5.3}$$

Les valeurs du profil croisé normalisé du processus $\mathbf{X}^{(i)}$ forment le vecteur $\mathbf{Q}^{(i)}_{\widehat{\mathcal{G}}}$ où

$$\left(\mathbf{Q}^{(i)}_{\widehat{\mathcal{G}}}\right)_j = Q_{\widehat{\mathcal{G}}}\left(\widehat{G}(\mathbf{X}^{(j)})|\mathbf{X}^{(i)}\right) \tag{5.4}$$

La figure 5.4 donne un exemple de profil croisé normalisé : ce profil est construit à partir du profil croisé présenté dans la figure 5.3.

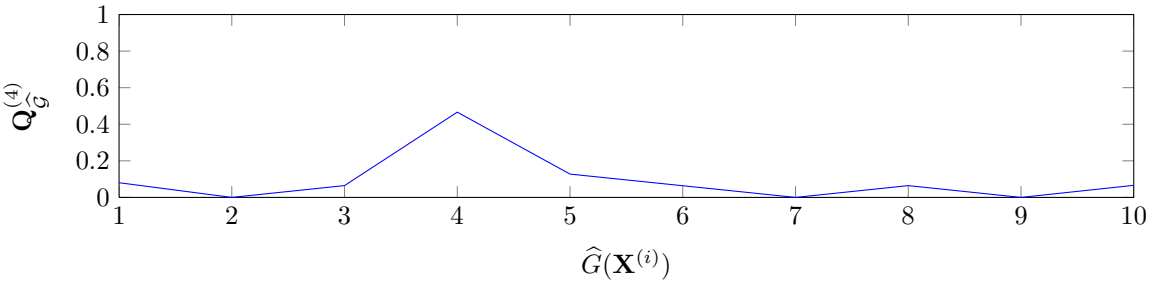



FIGURE *5.4 – Version normalisée du profil croisé présenté dans la figure 5.3.*

Pour chacun des processus nous obtenons un profil croisé normalisé qui peut être comparé à ceux des autres processus car ils sont construits sur le même ensemble de graphes et ils sont normalisés de façon à être assimilables à des profils de densité.

#### 5.2.2.3 Divergence de Kullback-Leibler symétrisée

Pour comparer les profils des différents processus, nous introduisons la divergence de Kullback-Leibler symétrisée. Elle est construite à partir de la divergence de Kullback-Leibler qui mesure la dissimilarité entre deux profils de densité.

**Définition 5.2.3.** ***Divergence de Kullback-Leibler symétrisée***

*Soient $P_1$ et $P_2$ deux profils de densité, la divergence de Kullback-Leibler symétrisée $\widetilde{D}_{KL}$ est définie ainsi :*

$$\widetilde{D}_{KL}(P_1, P_2) = D_{KL}(P_1|P_2) + D_{KL}(P_2|P_1) \tag{5.5}$$

*avec la divergence de Kullback-Leibler définie par :*

$$D_{KL}(P_1|P_2) = \sum_i P_1(i) \log\left(\frac{P_1(i)}{P_2(i)}\right). \tag{5.6}$$

La divergence de Kullback-Leibler symétrisée est une métrique symétrique, supérieure ou égale à zéro (dans le cas où les deux profils sont identiques) mais elle ne respecte pas l'inégalité triangulaire, ce n'est donc pas une distance. Elle informe sur la dissimilarité entre deux profils de densité : plus les profils sont différents, plus la divergence est élevée.

Comme les profils croisés normalisés introduits précédemment sont assimilables à des profils de densité, la divergence de Kullback-Leibler symétrisée est une métrique pertinente pour comparer ces profils.

Nous souhaitons par exemple comparer les trois profils croisés normalisés de la figure 5.5. Les profils sont construits sur le même ensemble de graphes (celui de la figure 5.2). La valeur de la divergence de Kullback-Leibler symétrisée entre chaque paire de profils est donnée pour les trois paires possibles.

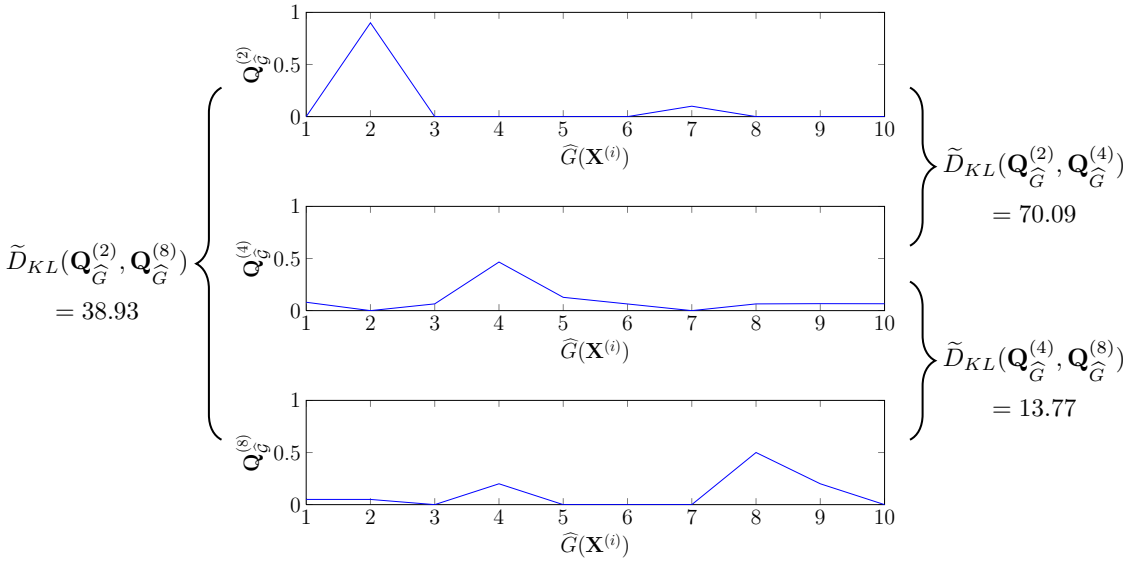



FIGURE 5.5 – *Profils croisés normalisés et valeurs de la divergence de Kullback-Leibler symétrisée pour chaque paire de profils.*

Les profils $\mathbf{Q}^{(2)}_{\widehat{\mathcal{G}}}$ et $\mathbf{Q}^{(4)}_{\widehat{\mathcal{G}}}$ sont très différents (les graphes pour lesquels l'un des profils a une valeur non nulle donnent une valeur nulle dans l'autre profil), ce sont eux qui ont la plus grande divergence de Kullback-Leibler symétrisée. Les profils $\mathbf{Q}^{(4)}_{\widehat{\mathcal{G}}}$ et $\mathbf{Q}^{(8)}_{\widehat{\mathcal{G}}}$ ont une allure assez similaire et leur divergence de Kullback-Leibler symétrisée est la plus faible des trois.

Nous avons introduit la notion de profils croisés normalisés permettant ainsi d'avoir des profils comparables pour chacun des $k$ processus que nous souhaitons comparer, ces profils étant assimilables à des profils de densité. De plus, nous avons introduit la divergence de Kullback-Leibler qui est une métrique permettant de comparer des profils de densité. Dans la suite, nous allons classifier les processus en utilisant soit directement leurs profils croisés normalisés soit la divergence de Kullback-Leibler symétrisée entre toutes les paires profils afin de séparer les processus en deux groupes.

## 5.3 Classification à l'aide des profils croisés normalisés

Considérons que nous sommes en présence de processus séparés en deux groupes, chaque groupe ayant pour structure d'indépendance conditionnelle une structure différant de celle de l'autre groupe de plusieurs arêtes. Nous souhaitons savoir dans quelle mesure nous pouvons différencier ces deux groupes, c'est-à-dire distinguer les structures d'indépendance conditionnelle. Notons que dans un groupe, tous les processus n'ont pas obligatoirement la même structure mais peuvent avoir des structures qui diffèrent d'un petit nombre d'arêtes.

Après avoir présenté succinctement la méthode de classification que nous utilisons, nous étudions les performances de classifications de groupes de processus simulés en utilisant les profils croisés normalisés et la divergence de Kullback-Leibler symétrisée sur ces mêmes profils. Dans un premier temps, nous présentons les résultats obtenus sur des groupes simples puis sur des groupes plus représentatifs de ce que peut être un groupe dans le cas de processus réels.

### 5.3.1 Classification par SVM

Pour faire la classification, nous utilisons une technique de machines à support de vecteurs (SVM pour Support Vector Machine en anglais) [CST00, HTF09]. Plus exactement nous utilisons les fonctions de Matlab® *svmtrain* et *svmclassify*. Ces fonctions sont utilisées avec les paramètres par défaut, c'est-à-dire :

- les données sont standardisées,
- le noyau est linéaire,
- l'algorithme utilisé pour trouver l'hyperplan de séparation est l'algorithme SMO (Sequential Minimal Optimization).

### 5.3.2 Procédures

Nous souhaitons classifier des processus appartenant à deux groupes de même taille, $k/2$. Nous prenons des groupes de taille $k/2$ car c'est la configuration la plus favorable pour faire de la classification et donc nous essayons d'être dans cette configuration même avec des données réelles. Pour réaliser la classification, nous allons utiliser deux entrées distinctes : les profils croisés normalisés et la divergence de Kullback-Leibler symétrisée entre les paires de profils. Dans la suite, nous comparons les performances de classification en fonction de l'entrée choisie.

En utilisant les profils croisés normalisés en entrée du classifieur, nous procédons de la manière suivante :

1. Nous estimons le graphe représentatif de la structure d'indépendance conditionnelle de chaque processus en utilisant la méthode ABiGlassoMaxLoop avec $step_\Lambda = 0.1$ et $e = 1$ pour chacun des $k$ processus. Nous obtenons l'ensemble $\widehat{\mathcal{G}}$.
2. Nous faisons la classification (étape que nous répétons $n_{it}$ fois avec des ensembles d'apprentissage différents) :
   (a) Nous sélectionnons aléatoirement $k_A/2$ processus de chaque groupe pour faire l'ensemble d'apprentissage $A$, les $k_T = k - k_A$ processus restant formant l'ensemble de test $T$.
   (b) Nous calculons le profil croisé pour chacun des $k$ processus sur l'ensemble $\widehat{\mathcal{G}}_A$ des graphes représentatifs des $k_A$ processus de l'ensemble d'apprentissage.
   (c) Nous normalisons les profils de scores croisés pour obtenir les profils croisés normalisés pour chacun des $k$ processus : pour $\mathbf{X}^{(i)}, (i) \in A \cup T$, nous obtenons $\mathbf{Q}^{(i)}_{\widehat{\mathcal{G}}_A}$. Notons que les profils, associés à un processus de l'ensemble d'apprentissage ou de test, sont calculés sur $\widehat{\mathcal{G}}_A$, ils sont donc normalisés sur ce même ensemble, c'est-à-dire sur l'ensemble d'apprentissage.
   (d) Nous apprenons l'ensemble d'apprentissage avec la fonction *svmtrain* et avec pour entrée les profils croisés normalisés des $k_A$ processus de l'ensemble d'apprentissage $\mathbf{Q}^{(i)}_{\widehat{\mathcal{G}}_A}, (i) \in A$.
   (e) Nous classifions l'ensemble de test avec la fonction *svmclassify* et avec pour entrée les profils croisés normalisés des $k_T$ processus de l'ensemble de test $\mathbf{Q}^{(i)}_{\widehat{\mathcal{G}}_A}, (i) \in T$.

(f) Nous calculons les performances de la classification obtenue : nous comptons combien de processus sont classés dans leur groupe d'origine pour les deux groupes utilisés sachant que nous utilisons toujours $k_T/2$ processus de chaque groupe pour la phase de test.

Pour la classification avec la divergence de Kullback-Leibler symétrisée, nous remplaçons les étapes 2.(d) et 2.(e) de la procédure décrite précédemment par :

(c).bis Nous calculons la divergence Kullback-Leibler symétrisée entre les profils croisés normalisés de tous les couples de processus : pour $\big((i),(j)\big) \in (A \cup T)^2$, nous obtenons $\widetilde{D}_{KL}\big(\mathbf{Q}^{(i)}_{\widehat{\mathcal{G}}_A}, \mathbf{Q}^{(j)}_{\widehat{\mathcal{G}}_A}\big)$.

(d) Nous apprenons l'ensemble d'apprentissage en utilisant la fonction *svmtrain* avec pour entrée les divergences de Kullback-Leibler symétrisées entre les $k_A$ processus de l'ensemble d'apprentissage, nous obtenons $\widetilde{D}_{KL}\big(\mathbf{Q}^{(i)}_{\widehat{\mathcal{G}}_A}, \mathbf{Q}^{(j)}_{\widehat{\mathcal{G}}_A}\big), \big((i),(j)\big) \in A^2$.

(e) Nous classifions l'ensemble de test en utilisant la fonction *svmclassify* avec pour entrée les divergences de Kullback-Leibler symétrisées entre, pour chaque processus de l'ensemble de test, le processus considéré et les $k_A$ processus de l'ensemble d'apprentissage, nous obtenons $\widetilde{D}_{KL}\big(\mathbf{Q}^{(i)}_{\widehat{\mathcal{G}}_A}, \mathbf{Q}^{(j)}_{\widehat{\mathcal{G}}_A}\big)$, $(i) \in T, (j) \in A$.

### 5.3.3 Métriques pour évaluer les performances de la classification

Pour évaluer de manière quantitative les performances de classification, nous assimilons notre classification à un problème de classification binaire : la condition positive revient à appartenir au deuxième groupe et la condition négative d'appartenir au premier. Nous rappelons que nous souhaitons discriminer des sujets sains de sujets malades d'où le choix d'une classification binaire. Nous pouvons ainsi utiliser la spécificité pour les performances de classification sur le premier groupe, la sensibilité pour les performances de classification sur le deuxième groupe et l'efficacité pour les performances globales. Les définitions de la spécificité et de la sensibilité sont données dans le chapitre 2, section 2.2.1.

**Définition 5.3.1.** ***Efficacité :***

*Pour deux groupes de données,*

$$eff = \frac{VN + VP}{VN + VP + FP + FN}$$

*où $VN$ désigne le nombre de processus du premier groupe bien classés, $VP$ le nombre de processus du second groupe bien classés, $FN$ le nombre de processus du second groupe classés dans le premier groupe et $FP$ le nombre de processus du premier groupe classés dans le second groupe.*

Si la classification est parfaite, l'efficacité vaut 1 et elle décroît au fur et à mesure que la classification se dégrade jusqu'à être nulle pour une classification où aucun élément n'est bien classifié.

### 5.3.4 Performances sur des données simulées

Nous souhaitons faire de la classification de données à l'aide des structures d'indépendance conditionnelle, c'est-à-dire regrouper les processus ayant des structures d'indépendance conditionnelle similaires en comparant les scores obtenus pour différentes structures et différents processus. Pour faire une telle classification, nous utilisons les procédures présentées dans la section précédente en prenant en entrée les profils croisés normalisés et les divergences de Kullback-Leibler symétrisées entre ces profils. Nous étudions les performances dans un premier temps sur des groupes de processus simulés simples puis sur des groupes de processus simulés plus réalistes

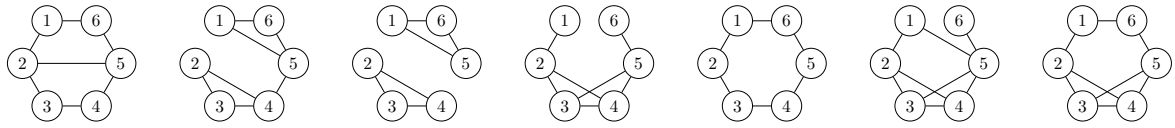
pour représenter des groupes de processus réels. Enfin, nous comparons les performances obtenues avec notre approche avec celles obtenues avec des métriques plus basiques. Les performances sont évaluées en utilisant la sensibilité, la spécificité et l'efficacité.

#### 5.3.4.1 Performances sur des données simulées simples

Dans un premier temps, nous travaillons avec des données simples, c'est-à-dire que nous cherchons à différencier deux groupes de processus dont la structure d'indépendance conditionnelle est la même au sein d'un même groupe.

**Données**

Nous générons sept groupes de processus, chacun associé à l'une de sept premières structures utilisées pour simuler les processus dans les chapitres 3 et 4. La figure 5.6 rappelle ces sept structures. Nous rappelons également que les structures 2, 3 et 4 sont décomposables.




(a) s1 (b) s2 (c) s3 (d) s4 (e) s5 (f) s6 (g) s7

FIGURE *5.6 – Rappel des structures utilisées pour générer des données dont la structure d'indépendance conditionnelle est connue.*

Le tableau 5.1 donne les couples de structures en fonction de leur distance de Hamming, c'est-à-dire le nombre d'arêtes de différence entre les deux structures du couple. La figure 5.7 donne la même information mais sous forme d'image en niveau de gris : une distance de Hamming nulle entre deux structures donne un pixel blanc et une distance de Hamming de 5 donne un pixel noir, les distances comprises entre ces deux valeurs s'échelonnent en niveau de gris.

| distance de Hamming | couple de structures ayant cette distance de Hamming |
|---|---|
| 1 | s1-s5 s2-s3 s4-s6 s4-s7 |
| 2 | s5-s7 s6-s7 |
| 3 | s1-s7 s2-s5 s2-s6 s2-s7 s4-s5 |
| 4 | s1-s2 s1-s4 s1-s6 s2-s4 s3-s5 |
| | s3-s6 s3-s7 s5-s6 |
| 5 | s1-s3 s3-s4 |

TABLEAU *5.1 – Couples de structures classé en fonction de leur distance de Hamming.*

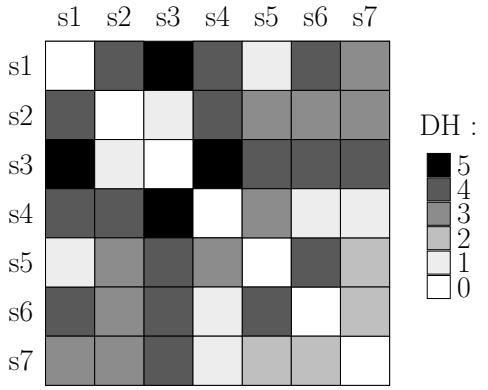



FIGURE *5.7 – Distance de Hamming en niveau de gris.*

Pour chaque groupe, nous prenons les dix matrices de covariance théoriques utilisées dans les chapitres 3 et 4, et nous générons un processus par matrice de covariance théorique pour chaque valeur de $n$.

Nous appliquons la procédure de classification sur les différents couples de groupes possibles pour évaluer les performances de cette procédure en fonction de la distance de Hamming entre les structures des deux groupes comparés. Nous faisons également varier le nombre d'observations

des processus pour observer l'influence du nombre d'observations sur les performances de la procédure. Nous choisissons de travailler avec $n \in \{12, 30, 60, 120, 300, 600\}$.

### Étude des métriques en entrée du classifieur

Pour mieux comprendre l'information contenue dans les entrées que nous avons choisi d'utiliser en entrée du classifieur, nous considérons le couple des groupes associés aux structures 1 et 3 (couple s1-s3) dont les structures sont distantes de 5 arêtes et le couple des groupes associés aux structures 1 et 5 (couple s1-s5) dont les structures sont distantes d'une seule arête. De plus, nous travaillons uniquement avec $n \in \{60, 600\}$.

**Entrée : profils croisés normalisés** La figure 5.8 illustre comment sont construites les matrices de la figure 5.9 qui représentent les profils croisés normalisés des $k = 20$ processus sur l'ensemble des $k_A = 16$ graphes de l'ensemble $\widehat{\mathcal{G}}_A$ pour $n = 60$ et $n = 600$. Les lignes correspondent aux profils et les colonnes aux graphes.

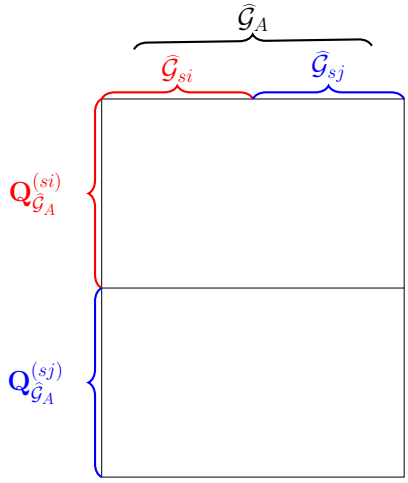
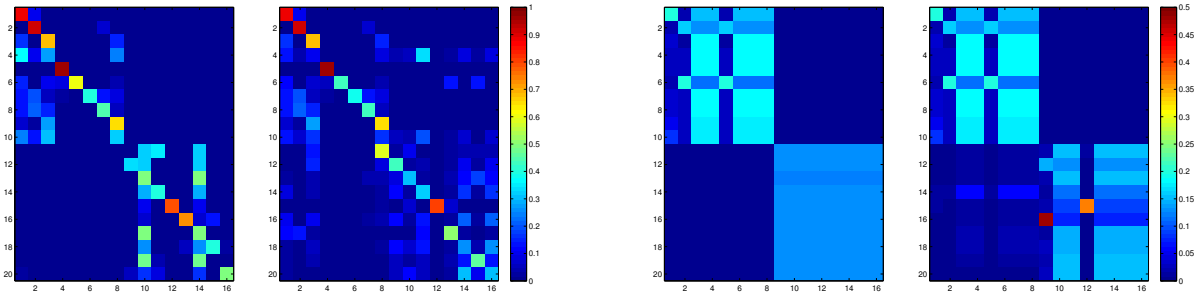



FIGURE 5.8 – *Illustration de la construction des matrices de la figure 5.9 : les colonnes correspondent aux structures d'indépendance conditionnelle estimées des processus de l'ensemble d'apprentissage et les lignes correspondent aux profils croisés normalisés sur $\widehat{\mathcal{G}}_A$. Les éléments en rouge sont ceux associés au premier groupe et en bleu ceux associés au second. Pour la figure 5.9, le premier groupe est celui associé à la structure 1 et le deuxième groupe est soit celui associé à la structure 3 soit celui associé à la structure 5.*



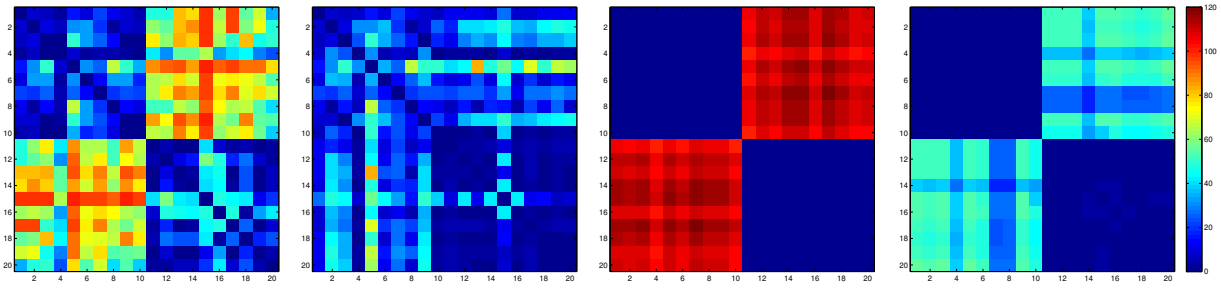
(a) s1-s3 $n = 60$ (b) s1-s5 $n = 60$ (c) s1-s3 $n = 600$ (d) s1-s5 $n = 600$

FIGURE 5.9 – *Profils croisés normalisés pour les couples s1-s3 (groupes associés aux structures 1 et 3) et s1-s5 (groupes associés aux structures 1 et 5) et pour $n = 60$ et $n = 600$. Pour $n = 600$ la distinction entre les deux groupes est visible nettement à l'œil nu, elle est moins nette pour $n = 60$. De plus, pour $n = 60$, la distinction est plus évidente pour le couple s1-s3 que pour le couple s1-s5.*

Pour $n = 600$, les groupes associés à différentes structures peuvent être distingués visuellement à partir des profils croisés normalisés des processus sur l'ensemble des structures d'indépendance conditionnelle estimées de l'ensemble d'apprentissage. Pour $n = 60$, la distinction est

moins nette. Cependant, la distinction est plus facile pour le couple s1-s3 que pour le couple s1-s5. En effet plus les structures utilisées pour générer les processus sont distantes en terme de distance de Hamming, moins la structure d'indépendance conditionnelle estimée d'un processus d'un des deux groupes sera représentative de la structure d'indépendance conditionnelle d'un processus de l'autre groupe.

**Entrée : divergences de Kullback-Leibler symétrisées** Pour la divergence de Kullback-Leibler symétrisée, les observations sont similaires. Les matrices regroupant les divergences de Kullback-Leibler symétrisées de toutes les paires de profils sont représentées dans la figure 5.10. Elles sont de taille $k \times k$ avec $k = 20$. La première moitié des indices correspond au premier groupe et la deuxième moitié au second groupe.



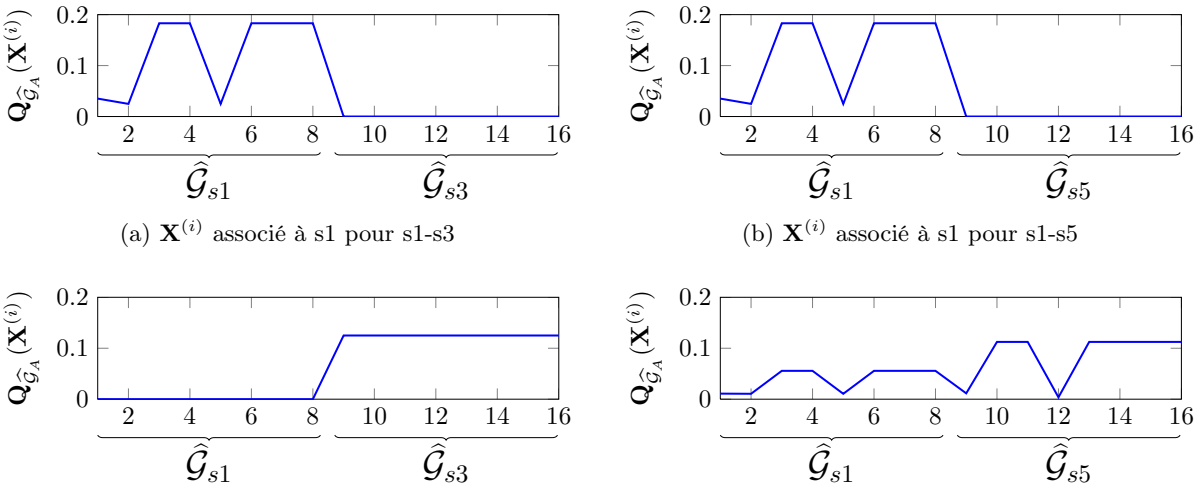
(a) s1-s3 $n = 60$ (b) s1-s5 $n = 60$ (c) s1-s3 $n = 600$ (d) s1-s5 $n = 600$

FIGURE *5.10 – Matrices des divergences de Kullback-Leibler symétrisées pour les couples s1-s3 et s1-s5 et* $n \in \{60, 600\}$*. Pour* $n = 600$*, la distinction est très nette, pour* $n = 60$*, elle l'est moins. De plus, pour* $n = 60$*, la distinction est plus évidente pour le couple s1-s3 que pour le coupe s1-s5.*

L'observation de la divergence de Kullback-Leibler symétrisée permet de constater que la distance de Hamming entre les structures utilisées pour générer les processus impacte les mesures utilisées pour la classification quel que soit $n$. Pour $n = 60$, quand les structures sont proches (couple s1-s5), la distinction entre les deux groupes n'est pas évidente visuellement, par contre elle est plus évidente dans le cas où les structures sont plus éloignées (couple s1-s3). Pour $n = 600$, la divergence entre un processus de chaque groupe est beaucoup plus élevée quand la distance de Hamming entre les structures des deux groupes est élevée.

La figure 5.11 permet de mieux comprendre l'information apportée par la divergence de Kullback-Leibler symétrisée. Elle représente quatre profils croisés normalisés, deux pour le couple s1-s3 (à gauche) et deux pour le couple s1-s5 (à droite). Prenons le profil du processus associé à la structure 5 (vignette (d)), les valeurs de $\mathbf{Q}_{\widehat{\mathcal{G}}_A}(\mathbf{X}^{(i)})$ sont élevées pour les graphes associés à la structure 5 mais sont également non nulles pour les graphes associés à la structure 1. Pour les autres profils, les valeurs du profil sont non nulles uniquement pour les graphes associés à la structure du processus. Comme les graphes associés à la structure 1 apportent de l'information sur les processus associés à la structure 5 et non sur les processus associés à la structure 3, la divergence de Kullback-Leibler symétrisée est plus faible entre les profils de processus associés aux structures 1 et 5 qu'aux structures 1 et 3.

Maintenant que nous avons vu l'allure des métriques en entrée du classifieur et l'information qu'elles contiennent, nous analysons les performances en sortie de ce classifieur.




(a) $\mathbf{X}^{(i)}$ associé à s1 pour s1-s3

(b) $\mathbf{X}^{(i)}$ associé à s1 pour s1-s5

(c) $\mathbf{X}^{(i)}$ associé à s3 pour s1-s3

(d) $\mathbf{X}^{(i)}$ associé à s5 pour s1-s5

FIGURE *5.11 – Profils croisés normalisés pour un processus de chaque groupe pour les couples s1-s3 et s1-s5 et* $n = 600$. *Ces profils permettent de mieux comprendre les valeurs de divergences de Kullback-Leibler symétrisées : plus deux profils sont proches, moins la divergence est élevée.*

### Étude des performances de classification

Pour étudier les performances de classification, nous considérons tous les couples possibles entre deux groupes associés à l'une des sept structures de la figure 5.6 et nous travaillons avec $n \in \{12, 30, 60, 120, 300, 600\}$.

**Entrée : profils croisés normalisés** En réordonnant la matrice des profils croisés normalisés pour que les premières lignes correspondent aux éléments de l'ensemble d'apprentissage, les $k_A = 16$ premières lignes sont utilisées pour faire l'étape d'apprentissage et les $k_T = 4$ dernières pour faire l'étape de test.

La figure 5.12 donne l'efficacité obtenue en faisant la classification à partir des profils croisés normalisés pour différentes valeurs du nombre d'observations : $n \in \{12; 30, 60, 120, 300, 600\}$. Pour chaque valeur de $n$, l'efficacité est moyennée sur les couples ayant la même distance de Hamming entre leurs structures. Deux phénomènes sont observés :

- pour une distance de Hamming donnée, plus $n$ est grand, plus l'efficacité tend vers 1
- pour $n$ fixé, plus la distance de Hamming entre les structures associées aux deux groupes est grande, plus l'efficacité tend vers 1.

Notons que pour $n = 600$, quelle que soit la distance de Hamming, l'efficacité est très proche voire égale à 1 et que pour $n = 60$ il est plus difficile de séparer deux groupes dont les structures sont proches que deux groupes dont les structures sont éloignées.

**Entrée : divergences de Kullback-Leibler symétrisées** Pour être utilisée pour la classification, la matrice des divergences de Kullback-Leibler symétrisées est réorganisée de façon à ce que les premiers indices (lignes et colonnes) correspondent aux éléments de l'ensemble d'apprentissage. La figure 5.13 illustre quelles parties de la matrice sont utilisées pour les différentes étapes de la classification.

La figure 5.14 donne l'efficacité obtenue en faisant la classification à partir des divergences de Kullback-Leibler symétrisées entre les profils croisés normalisés pour différentes valeurs du nombre d'observations : $n \in \{12, 30, 60, 120, 300, 600\}$. Pour chaque valeur de $n$, l'efficacité est

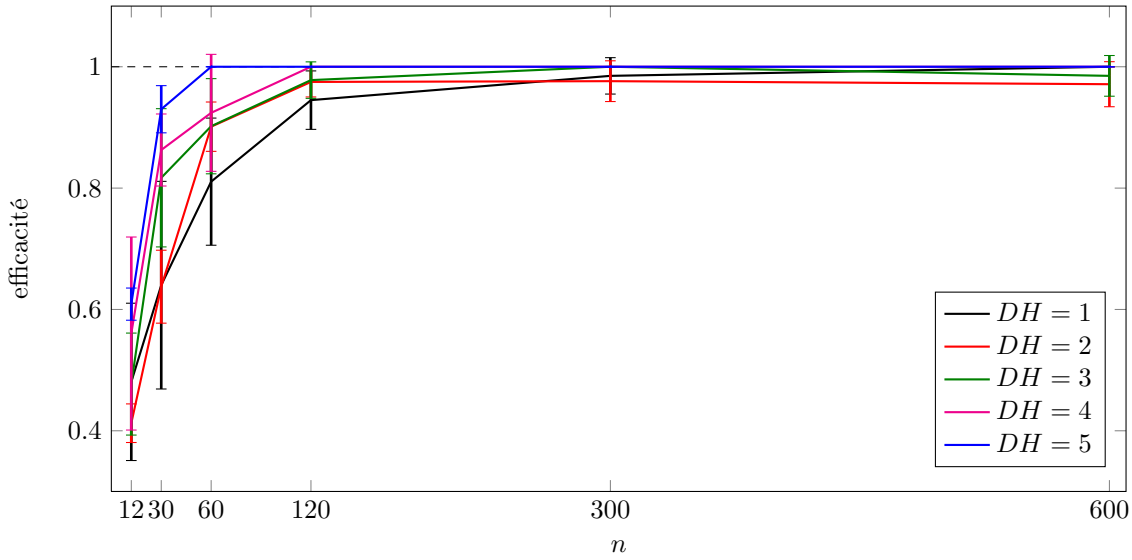



FIGURE 5.12 – *Efficacité de la classification en utilisant les profils croisés normalisés en entrée du classifieur. L'efficacité est moyennée sur les couples ayant la même distance de Hamming (DH) pour chaque valeur de* $n \in \{12, 30, 60, 120, 300, 600\}$. *Les barres d'erreur représentent l'écart-type.*

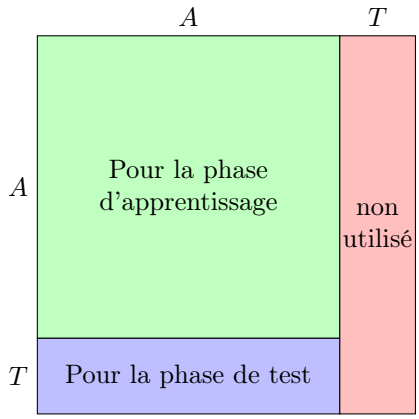



FIGURE 5.13 – *En réarrangeant les lignes et les colonnes de la matrice des divergences de Kullback-Leibler symétrisées de façon à ce que les premiers indices correspondent aux processus de l'ensemble d'apprentissage* $A$ *et les derniers aux processus de l'ensemble de test* $T$*, la figure permet de voir quelles parties de la matrice sont utilisées et à quel moment. Rappelons que la matrice est calculée pour* $\widehat{\mathcal{G}}_A$.

moyennée sur les couples ayant la même distance de Hamming entre leurs structures. Nous observons les mêmes phénomènes que ceux observés quand les profils croisés normalisés sont utilisés en entrée du classifieur. Par contre, pour une valeur du nombre d'observations donnée, les performances de classifications sont meilleures en utilisant la divergence de Kullback-Leibler symétrisée (l'efficacité est plus proche de 1).

Pour $n = 600$, l'efficacité vaut 1 quelle que soit la distance de Hamming entre les structures du groupes. Pour $n = 60$, il est plus facile de séparer deux groupes dont la distance de Hamming entre les structures est faible qu'avec les profils croisés normalisés en entrée du classifieur. Mais toujours pour $n = 60$, plus la distance de Hamming est élevée, plus il est facile de séparer les deux groupes.

Sur ces données simples, nous avons observé que :
- plus la distance de Hamming entre les structures des deux groupes est grande, plus il est facile de séparer ces deux groupes,
- plus le nombre d'observations est important, meilleures sont les performances de classification,
- l'utilisation de la divergence de Kullback-Leibler symétrisée entre les profils croisés normali-

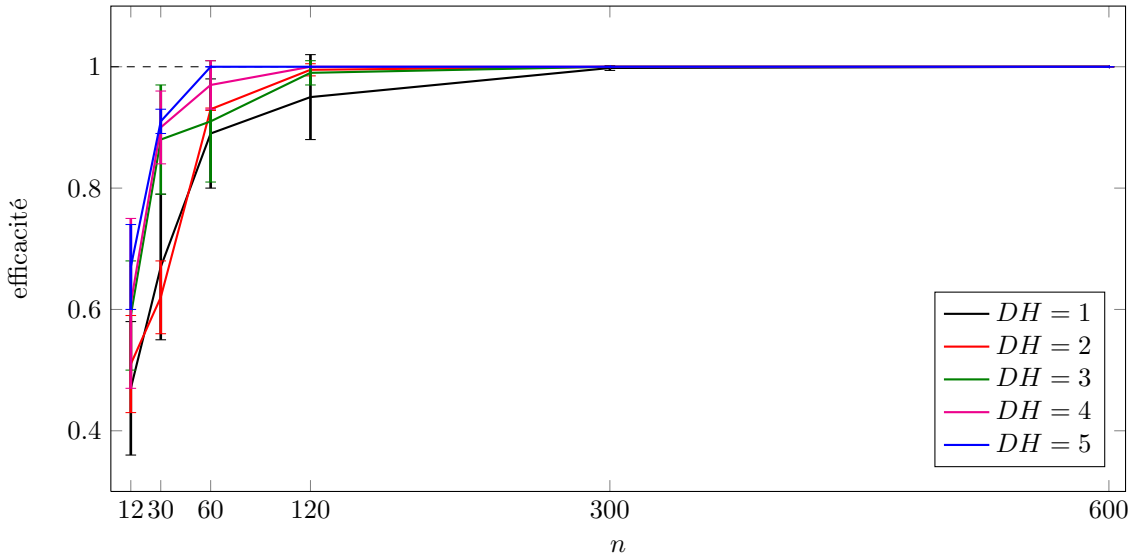



FIGURE *5.14 – Efficacité de la classification en utilisant la divergence de Kullback-Leibler symétrisée entre les profils croisés normalisés en entrée du classifieur. L'efficacité est moyennée sur les couples ayant la même distance de Hamming (DH) pour chaque valeur de* $n \in \{12, 30, 60, 120, 300, 600\}$. *Les barres d'erreur représentent l'écart-type.*

sés donne de meilleures performances que l'utilisation directe des profils croisés normalisés.

#### 5.3.4.2 Performances sur des données simulées réalistes

Si nous souhaitons séparer des données réelles, il n'est pas réaliste de considérer qu'une structure d'indépendance conditionnelle est associée à un groupe et une autre structure à l'autre groupe. Dans cette partie, nous travaillons avec des groupes de processus plus complexes que ceux de la partie précédente.

**Données**

Nous souhaitons faire de la classification entre des groupes de processus dont les structures d'indépendance conditionnelle diffèrent, entre les groupes, de plusieurs arêtes. Pour simuler du bruit au sein d'un groupe, nous considérons que les processus au sein d'un même groupe peuvent avoir des structures d'indépendance conditionnelle qui diffèrent d'une à deux arêtes maximum.

A partir des processus simulés pour étudier les performances des méthodes ABiGlasso, nous construisons différents groupes :

1. groupe 1 : 9 processus générés à partir de la structure 1 et 9 à partir de la structure 5. Les deux structures utilisées diffèrent d'une arête.
2. groupe 2 : 9 processus générés à partir de la structure 2 et 9 à partir de la structure 3. Les deux structures utilisées diffèrent d'une arête.
3. groupe 3 : 6 processus générés à partir de la structure 4, 6 à partir de la structure 6 et 6 à partir de la structure 7. Les structures 6 et 7 diffèrent de la structure 4 d'une arête.

La figure 5.15 rappelle les structures mentionnées ci-dessus pour chacun des trois groupes.

Le tableau 5.2 donne les distances de Hamming entre les différentes structures utilisées pour construire les groupes. Pour rappel, la distance de Hamming entre deux graphes donne le nombre d'arêtes qui diffèrent entre ces deux graphes.

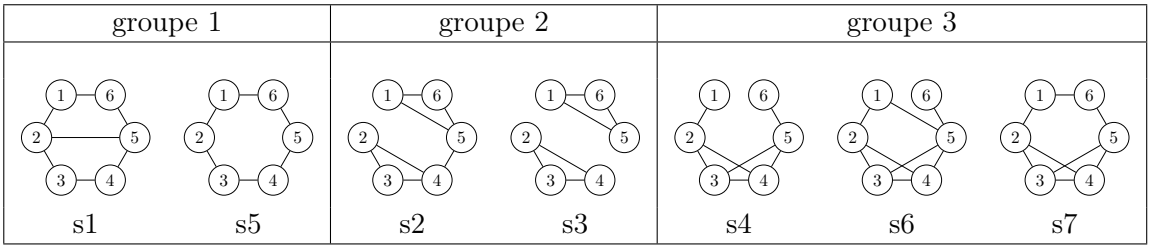



FIGURE *5.15 – Structures utilisées pour simuler les processus des trois groupes que nous allons classifier.*

| | s1 | s5 | s2 | s3 | s4 | s6 | s7 |
|---|---|---|---|---|---|---|---|
| s1 | 0 | **1** | 4 | 5 | 4 | 4 | 3 |
| s5 | | 0 | 3 | 4 | 3 | 4 | 2 |
| s2 | | | 0 | **1** | 4 | 3 | 3 |
| s3 | | | | 0 | 5 | 4 | 4 |
| s4 | | | | | 0 | **1** | **1** |
| s6 | | | | | | 0 | **2** |
| s7 | | | | | | | 0 |

TABLEAU *5.2 – Matrice des distances de Hamming entre structures. En rouge la distance de Hamming entre les structures du groupe 1, en bleu la distance de Hamming entre les structures du groupe 2 et en vert la distance de Hamming entre les structures du groupe 3.*

Notons que toutes les distances de Hamming entre des structures n'appartenant pas à un même groupe sont supérieures ou égales à 3 sauf entre la structure 5 et la structure 7 où la distance de Hamming vaut 2, comme entre les structures 6 et 7 qui appartiennent au même groupe. Nous verrons par la suite si cela impacte les performances de classification.

### Étude des métriques en entrée du classifieur

Nous étudions dans un premier temps les deux entrées sur lesquelles nous avons choisi d'utiliser le classifieur. Pour cela, nous considérons le couple groupe 1/groupe 3. Nous avons retenu ce couple pour illustrer notre propos mais des résultats similaires sont observés sur d'autres couples. Nous sommes en présence de $k = 36$ processus à classifier. Nous supposons que l'ensemble d'apprentissage est composé des 9 processus associés à la structure 1, de 6 processus associés à la structure 5, des 6 processus associés à la structure 4, des 6 processus associés à la structure 6 et de 3 processus associés à la structure 7. Nous avons donc $k_A = 30$ et $k_T = 6$.

La figure 5.16 illustre comment sont construites les matrices de la figure 5.17. Pour $n = 60$ et $n = 600$, chaque ligne des matrices de la figure 5.17 correspond au profil croisé normalisé d'un processus du couple groupe 1/groupe 3 sur les $k_4 = 30$ structures d'indépendance conditionnelle estimées des processus de l'ensemble d'apprentissage. La matrice est donc de taille $k \times k_A$ avec $k = 36$ et $k_A = 30$. La figure 5.16 illustre la façon dont sont construites les deux matrices de la figure 5.17.

Pour $n = 60$, il n'est pas très évident de dissocier à l'œil les deux groupes à partir des profils croisés normalisés. Par contre, pour $n = 600$, la matrice des profils est visuellement composée de blocs correspondant aux différents groupes.

Pour le même couple groupe 1/groupe 3, la figure 5.18 représente, pour $n = 60$ et $n = 600$, la matrice contenant la divergence de Kullback-Leibler symétrisée entre les profils présentés dans

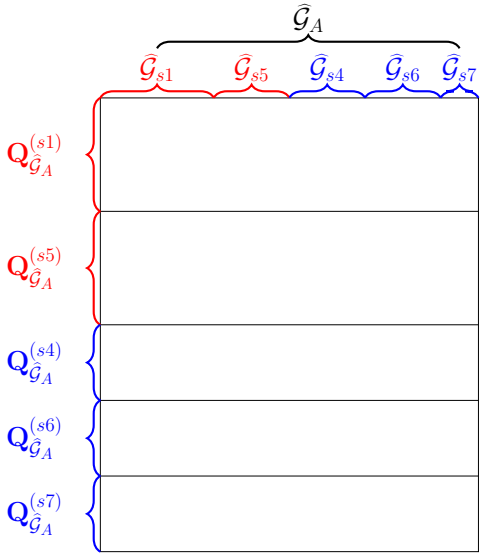
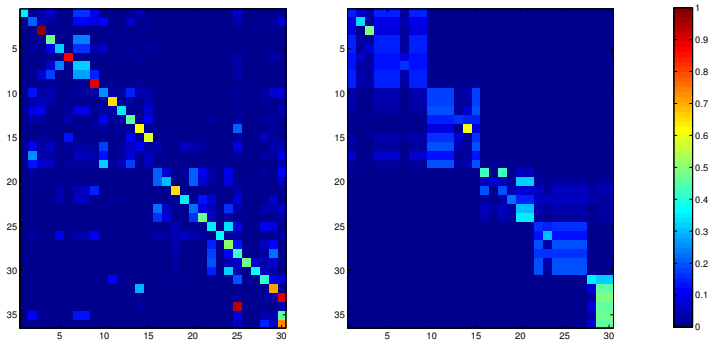


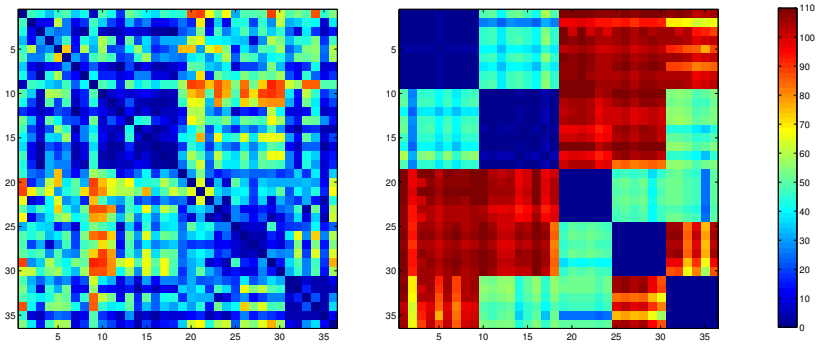

FIGURE *5.16 – Illustration de la construction des matrices de la figure 5.17 : les colonnes sont les structures d'indépendances estimées des processus de l'ensemble d'apprentissage et les lignes sont les profils croisés normalisés pour chaque processus du couple groupe 1/groupe 3. Les éléments associés au groupe 1 sont en rouge et ceux associés au groupe 3 sont en bleu et sx fait référence aux éléments associés à la structure x.*



(a) $n = 60$ (b) $n = 600$

FIGURE *5.17 – Matrices des profils croisés normalisés pour le couple groupe 1/groupe 3. Les premiers indices correspondent aux éléments du groupe 1 et les derniers aux éléments du groupe 3. Pour $n = 600$, les deux groupes peuvent être dissociés en regardant leurs profils croisés normalisés.*

la figure 5.17.

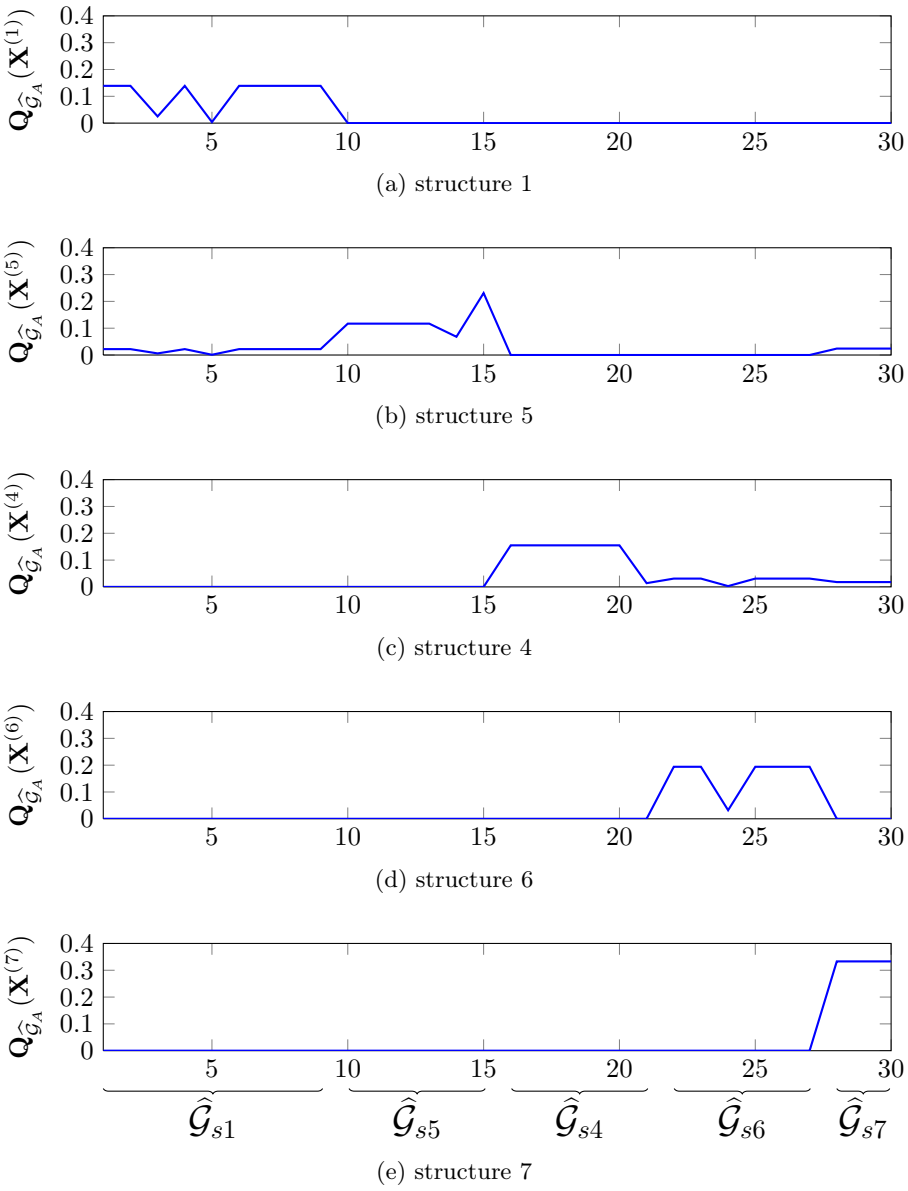


(a) $n = 60$ (b) $n = 600$

FIGURE *5.18 – Matrices des divergences de Kullback-Leibler symétrisées pour le couple groupe 1/groupe 3. Les premiers indices correspondent aux éléments du groupe 1 et les derniers aux éléments du groupe 3. Pour $n = 600$, les deux groupes peuvent être dissociés à l'œil.*

De même que pour les profils, les données peuvent être classées aisément pour $n = 600$ et moins facilement pour $n = 60$.

Notons que la divergence de Kullback-Leibler symétrisée est très proche de zéro quand les deux profils considérés sont ceux de processus générés à partir de la même structure et qu'elle augmente comme le nombre d'arêtes de différence entre les structures utilisées pour générer les processus augmente. Remarquons qu'entre les structures 5 et 7, de groupes distincts, la divergence de Kullback-Leibler symétrisée est plus faible qu'entre les structures 6 et 7, du même groupe, alors que dans les deux cas la distance de Hamming vaut 2. Nous allons essayer de comprendre pourquoi dans la suite.




(a) structure 1

(b) structure 5

(c) structure 4

(d) structure 6

(e) structure 7

FIGURE *5.19 – Profils croisés normalisés pour cinq processus générés à partir des cinq structures utilisées pour construire le couple groupe 1/groupe 3.*

La figure 5.19 permet de mieux comprendre l'information apportée par la divergence de Kullback-Leibler symétrisée. Elle représente cinq profils de densité de croisés, chacun pour une des cinq structures utilisées pour la construction des couples groupe 1/groupe 3. Prenons par

exemple le profil associé à la structure 5, les valeurs de $\mathbf{Q}_{\widehat{\mathcal{G}}_A}(\mathbf{X}^{(i)})$ sont élevées pour les graphes associés à la structure 5, faibles mais non nuls pour les graphes associés aux structures 1 et 7 et nuls pour les autres. Comparons maintenant aux valeurs de la divergence de Kullback-Leibler symétrisée : entre les profils associés à la structure 5, les valeurs sont quasiment nulles, entre les profils associés à la structure 5 et ceux associés aux structures 1 et 7, les valeurs sont moyennes (autour de 50) et entre les profils associés à la structure 5 et ceux associés aux structures où le profil est nul, les valeurs de la divergence de Kullback-Leibler symétrisée sont importantes (autour de 100).

Pour en revenir au fait que la divergence de Kullback-Leibler symétrisée est plus faible entre les profils associés aux structures 5 et 7 (inter groupes) que entre les profils associés aux structures 6 et 7 (intra-groupe), la figure 5.19 montre que le profil associé à la structure 5 a des valeurs non nulles pour les graphes associés à la structure 7 alors que ni le profil associé à la structure 6 ni le profil associé à la structure 7 n'ont des valeurs non nulles pour les graphes associés respectivement à la structure 7 et à la structure 6. Cela signifie que, bien que les structures 5 et 7 ne sont pas associées au même groupe, les graphes associés à la structure 7 apportent de l'information sur les processus simulés à partir de la structure 5 alors qu'ils n'en apportent pas sur les processus simulés à partir de la structure 6.

En regardant ces 5 profils, nous remarquons que ce sont les structures estimées ayant plus d'arêtes que la structure d'indépendance conditionnelle attendue qui portent de l'information sur la structure attendue : pour le profil associé à la structure 4, les valeurs pour les graphes associés aux structures 6 et 7 sont non nulles alors que l'inverse est faux. En effet, un graphe contenant la structure attendue contient toute l'information plus quelques arêtes assimilables à du bruit, son score est donc plus faible que celui de la structure attendue mais n'est pas nul : ce graphe est représentatif du processus étudié. Par contre, un graphe ayant même une unique arête en moins par rapport à la structure d'indépendance conditionnelle attendue ne contient pas toute l'information et a donc de grande chance d'avoir un score faible : il n'est pas assez représentatif du processus étudié.

Maintenant que nous avons vu l'allure des métriques en entrée du classifieur et l'information qu'elles contiennent dans le cas de groupes de processus simulés plus réalistes, nous analysons les performances en sortie de ce classifieur.

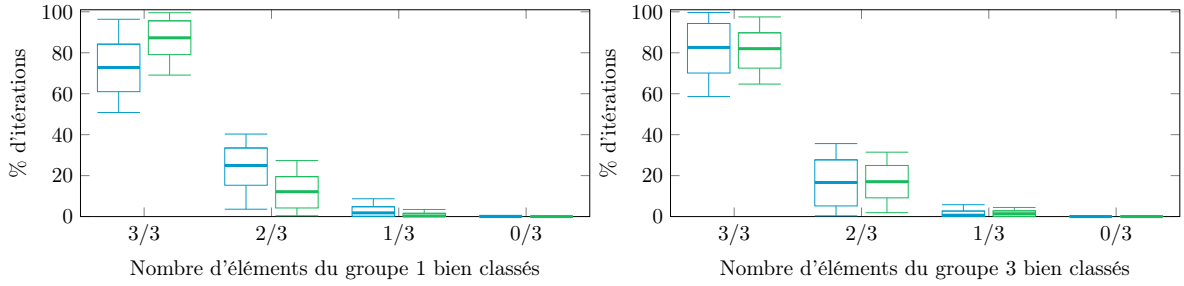
### Étude des performances de classification

En réordonnant la matrice des profils croisés normalisés pour que les premières lignes correspondent aux éléments de l'ensemble d'apprentissage, les $k_A = 30$ premières lignes sont utilisées pour faire l'étape d'apprentissage et les $k_T = 6$ dernières pour faire l'étape de test.

Concernant la matrice des divergences de Kullback-Leibler symétrisées, elle est réorganisée de façon à ce que les premiers indices (lignes et colonnes) correspondent aux éléments de l'ensemble d'apprentissage. La figure 5.13 illustre quelles parties de la matrice sont utilisées pour la classification à partir de la matrice des divergences de Kullback-Leibler symétrisées.

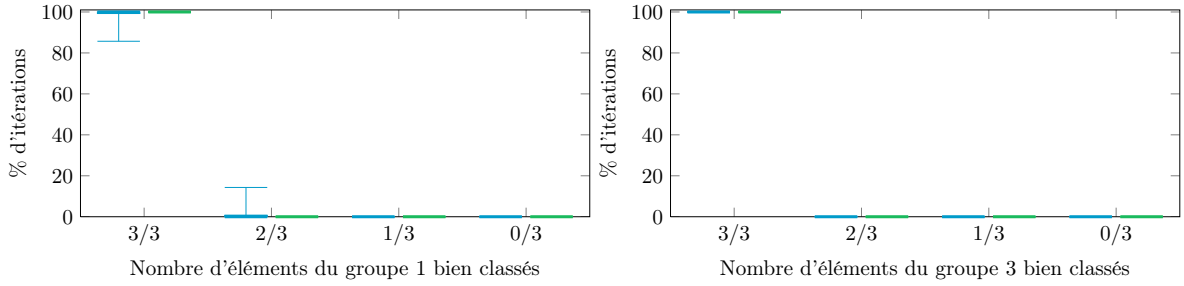
Nous faisons l'étude sur 500 couples de processus venant des groupes 1 et 3 et pour chaque couple, nous répétons l'étape de classification $n_{it} = 1000$ fois avec à chaque itération un nouvel ensemble d'apprentissage. La figure 5.20 illustre le pourcentage d'itérations de classification classifiant correctement $x/y$ processus pour un couple donné en faisant la classification directement à partir des profils croisés normalisés (en bleu) ou à partir de la divergence de Kullback-Leibler symétrisée entre ces profils (en vert), $y = 3$ et $x \in [\![0, 3]\!]$. Les boîtes à moustaches sont construites sur les 500 couples étudiés.

Pour $n = 600$, la classification est quasiment parfaite quelle que soit la métrique utilisée en entrée du classifieur et quel que soit le groupe considéré. Pour $n = 60$, les résultats sont




(a) $n = 60$, classification groupe 1

(b) $n = 60$, classification groupe 3




(c) $n = 600$, classification groupe 1

(d) $n = 600$, classification groupe 3

FIGURE *5.20 – Pourcentage d'itérations de classification classifiant correctement $x/y$ processus pour 500 couples, $y = 3$ et $x \in [\![0, 3]\!]$. Les entrées du classifieur sont les profils croisés normalisés (en bleu) et la divergence de Kullback-Leibler symétrisée entre ces profils (en vert).*

bons : pour plus de la moitié des couples, plus de 75% des itérations classifient correctement la totalité des éléments de l'ensemble de test. Cependant, les résultats concernant la classification des éléments du groupe 1 sont meilleurs quand la métrique utilisée est la divergence de Kullback-Leibler : la médiane sur les 500 couples vaut 72.8 pour les profils croisés normalisés en entrée du classifieur contre 87.3 pour la divergence de Kullback-Leibler symétrisée.

Les valeurs de sensibilité, de spécificité et d'efficacité pour les classifications avec en entrée soit les profils croisés normalisés soit la divergence de Kullback-Leibler symétrisée entre ces profils sont données dans le tableau 5.3.

| | $\mathbf{Q}_{\widehat{\mathcal{G}}_A}$ | | $\widetilde{D}_{KL}$ | |
|---|---|---|---|---|
| | $n = 60$ | $n = 600$ | $n = 60$ | $n = 600$ |
| spécificité (gp1) | 0.8975 | 0.9911 | 0.9484 | 0.9999 |
| sensibilité (gp3) | 0.9265 | 0.9998 | 0.9288 | 1 |
| efficacité | 0.912 | 09955 | 0.9386 | 1 |

TABLEAU *5.3 – Performances de la classification pour $n = 60$ et $n = 600$ avec pour entrée du classifieur soit les profils croisés normalisés $\mathbf{Q}_{\widehat{\mathcal{G}}_A}$ soit la divergence de Kullback-Leibler symétrisée $\widetilde{D}_{KL}$ sur les 500 couples de données groupe 1/groupe 3 et les 1000 itérations de classification.*

La même étude est menée sur des couples groupe 2/groupe 3. Les résultats sont donnés dans le tableau 5.4. Les résultats sont meilleurs que pour les couples groupe 1/groupe 3. Cela s'explique par le fait que les distances de Hamming entre les structures du groupe 2 et celles du groupe 3 sont plus grandes que celles entre les structures du groupe 1 et celles du groupe 3. Nous

| | $\mathbf{Q}_{\widehat{\mathcal{G}}_A}$ | | $\widetilde{D}_{KL}$ | |
|---|---|---|---|---|
| | $n = 60$ | $n = 600$ | $n = 60$ | $n = 600$ |
| spécificité (gp2) | 0.9236 | 1 | 0.9578 | 1 |
| sensibilité (gp3) | 0.9689 | 1 | 0.9811 | 1 |
| efficacité | 0.9463 | 1 | 0.9695 | 1 |

TABLEAU *5.4 – Performances de la classification pour $n = 60$ et $n = 600$ avec pour entrée du classifieur soit les profils croisés normalisés $\mathbf{Q}_{\widehat{\mathcal{G}}_A}$ soit la divergence de Kullback-Leibler symétrisée $\widetilde{D}_{KL}$ sur les 500 couples de données groupe 2/groupe 3 et les 1000 itérations de classification.*

avons vu lors de l'étude sur des groupes de processus simples que plus la distance de Hamming entre les structures des deux groupes étudiés était grande, plus il était facile de bien classifier ces deux groupes. C'est ce phénomène qui explique les meilleures performances de classification sur les couples groupe 2/groupe 3 que sur les couples groupe 1/groupe 3.

Dans tous les cas, la divergence de Kullback-Leibler symétrisée en entrée du classifieur donne de meilleurs résultats que l'utilisation directe des profils croisés normalisés.

#### 5.3.4.3 Comparaison avec d'autres métriques plus basiques

Nous proposons une procédure de classification de groupes de processus basée sur les scores des différentes estimées de la structure d'indépendance conditionnelle des processus étudiée. Pour construire cette procédure, nous avons introduit une nouvelle métrique : la divergence de Kullback-Leibler symétrisée. Nous comparons notre approche à des approches plus simples, aussi bien dans leur formulation que dans leur mise en œuvre afin d'étudier si la complexité de notre procédure se justifie par l'obtention de meilleures performances.

Une métrique basique utilisée pour classifier des processus est la norme de Frobenius de la différence des matrices de corrélation. Une autre métrique prenant en compte une approche par modèles graphiques gaussiens est la norme de Frobenius de la différence des matrices de corrélation estimée à partir de la méthode BAGlasso (Bayesian Adaptive Graphical lasso). Nous comparons les performances entre l'utilisation de ces deux métriques et l'utilisation de la divergence de Kullback-Leibler symétrisée sur les 500 couples de processus groupe 1/groupe 3. Les valeurs de l'efficacité pour les deux entrées sont données dans le tableau 5.5.

| | $\|\|M_i - M_j\|\|_{\mathrm{F}}, M = \widehat{C}_{\mathrm{emp}}$ | | $\|\|M_i - M_j\|\|_{\mathrm{F}}, M = \widehat{C}_{\mathrm{BAG}}$ | | $\widetilde{D}_{KL}$ | |
|---|---|---|---|---|---|---|
| | $n = 60$ | $n = 600$ | $n = 60$ | $n = 600$ | $n = 60$ | $n = 600$ |
| spécificité (gp1) | 0.49048 | 0.4884 | 0.52213 | 0.629 | 0.9696 | 0.99988 |
| sensibilité (gp3) | 0.47504 | 0.47289 | 0.58081 | 0.74378 | 0.92878 | 1 |
| efficacité | 0.48276 | 0.48065 | 0.55147 | 0.68639 | 0.94919 | 0.99999 |

TABLEAU *5.5 – Performances de la classification pour $n = 60$ et $n = 600$ avec pour entrée du classifieur soit la norme de Frobenius de la différence des matrices de corrélation empiriques $||M_i - M_j||_F, M = \widehat{C}_{emp}$ soit la norme de Frobenius des matrices de corrélation estimée à partir de la méthode BAGlasso $||M_i - M_j||_F, M = \widehat{C}_{BAG}$ soit la divergence de Kullback-Leibler symétrisée $\widetilde{D}_{KL}$ sur les 500 couples de données groupe 1/groupe 3 et 500 itérations de classification.*

Quel que soit $n$, l'utilisation de la norme de Frobenius sur la différence des matrices de corrélation donne des résultats dont l'efficacité est similaire à une classification aléatoire (efficacité de 0.5). L'utilisation de la norme de Frobenius sur la différence des matrices de corrélation estimée par la méthode BAGlasso donne de meilleurs résultats : l'efficacité atteint 0.69 pour $n = 600$. L'utilisation de la divergence de Kullback-Leibler symétrisée sur les profils croisés normalisés des processus comparés donne des résultats nettement meilleurs que les deux autres approches, l'efficacité étant supérieure à 0.94 même pour $n = 60$.

## 5.4 Discussion

Dans ce chapitre, nous avons présenté une méthode pour faire de la comparaison de processus en utilisant les estimées de leurs structures d'indépendance conditionnelle ainsi que leurs scores. Ne pouvant utiliser les métriques de graphes usuelles puisque nous sommes en présence de graphes avec un nombre d'arêtes variable, nous choisissons de construire des profils croisés normalisés : ces profils sont construits sur l'espace des estimées des structures d'indépendance conditionnelle à l'aide du score $z$, introduit lors de la création de la méthode ABiGlasso, et normalisés pour être assimilables à des profils de densité. Les dissimilarités entre ces profils sont ensuite mesurées à l'aide de la divergence de Kullback-Leibler symétrisée. Nous utilisons cette métrique pour faire de la classification via SVM (support vector machine). Nous avons vu que même dans la configuration $n > p$ (ici $n = 60$ et $p = 6$), configuration pour laquelle l'estimation de la structure d'indépendance conditionnelle n'est pas toujours de très bonne qualité, les résultats de classification utilisant la divergence de Kullback-Leibler symétrisée sur les profils croisés normalisés sont assez bons. Cette approche ouvre des perspectives intéressantes dans le cas de l'étude de processus réels. Nous proposons une application de cette approche à des données neurologiques dans le chapitre suivant.

Lors des études des processus croisés normalisés, que ce soit sur des données simples ou sur des données plus réalistes, nous avons constaté que les scores sont non nuls soit si la structure estimée est la structure attendue, soit si la structure estimée contient la structure attendue. Ceci n'est qu'une observation mais c'est un point important pour des études ultérieures : si toute structure ayant un score non nul contient la structure attendue, cela implique que la structure attendue est contenue dans l'intersection des différentes structures ayant un score non nul.

# Chapitre 6

# Application dans le contexte de la connectivité fonctionnelle cérébrale

**Sommaire**



Dans ce chapitre, nous nous intéressons à l'étude la connectivité fonctionnelle au sein du cerveau. L'étude de la connectivité fonctionnelle cérébrale est l'étude des interactions entre l'activation neuronale de différentes parties du cerveau. Une connexion fonctionnelle existe quand deux zones du cerveau s'activent simultanément ou l'une après l'autre. L'étude de la connectivité fonctionnelle peut se faire à partir de différents types de données mais nous nous intéressons uniquement aux données d'IRM (imagerie par résonance magnétique) fonctionnelle. Le temps d'acquisition entre deux échantillons d'un signal d'IRM fonctionnelle étant grand devant les phénomènes neuronaux, les études de la connectivité fonctionnelle à partir de ce type de données se concentrent uniquement sur les connexions instantanées.

Dans le cadre de cette thèse, nous abordons l'étude de la connectivité fonctionnelle comme l'étude de la structure d'indépendance conditionnelle d'ensembles de régions cérébrales. Cette approche n'est pas courante et peu de résultats ont été obtenus en l'utilisant. L'objectif de notre travail est de mettre en lumière des ensembles de régions cérébrales permettant d'étudier

comment se différencie la connectivité des patients dans le coma de celle des sujets sains en appliquant la méthode présentée dans le chapitre 5.

Dans une première partie, nous présentons le principe de l'IRM fonctionnelle, le format des données et comment ces données sont utilisées pour étudier la connectivité fonctionnelle. Dans une deuxième partie, après avoir présenté les données avec lesquelles nous travaillons, nous appliquons la méthode introduite dans le chapitre 5 pour trouver des ensembles de régions cérébrales qui permettent de différencier la connectivité fonctionnelle des patients dans le coma de celle de sujets sains et qui sont donc pertinents pour étudier l'activité cérébrale de ces patients.

## 6.1 IRM fonctionnelle et connectivité fonctionnelle

L'IRM fonctionnelle et l'étude de la connectivité fonctionnelle cérébrale sont des domaines de recherches apparus dans les années 90 et en plein essor (environ 426 000 résultats sur google scholar pour *fMRI* sans renseigner de date et environ 16 300 résultats depuis 2014 [1]). Nous allons voir dans cette partie ce qu'est l'IRM fonctionnelle et en quoi consiste l'étude de la connectivité fonctionnelle à partir de données d'IRM fonctionnelle.

### 6.1.1 IRM fonctionnelle

L'IRM (imagerie par résonance magnétique) est une technique d'imagerie médicale non invasive qui génère des images à partir de l'enregistrement de variations de champs magnétiques. Elle peut être utilisée pour étudier différentes parties du corps humain comme le cerveau ou le cœur. Elle est principalement connue pour fournir des images précises de la structure de l'organe étudié. Dans le cadre de l'étude du cerveau, l'IRM peut également être utilisée pour étudier l'activité cérébrale : on parle alors d'IRM fonctionnelle. L'objectif de la modalité IRM fonctionnelle est d'étudier comment les différentes parties du cerveau interagissent entre elles.

#### 6.1.1.1 Principe

L'IRM fonctionnelle enregistre l'activité neuronale en enregistrant le phénomène métabolique lié à cette activité : lorsqu'un neurone entre en activité, il consomme de l'oxygène. Cette consommation d'oxygène est compensée, même surcompensée, par une augmentation locale du flux sanguin. L'augmentation du flux sanguin implique une augmentation de la quantité d'hémoglobine oxygénée et donc une diminution de la concentration en hémoglobine désoxygénée. Les variations de la concentration en hémoglobine désoxygénée forment le signal BOLD (Blood Oxygen Level Dependent). C'est ce signal qui est mesuré en IRM fonctionnelle. Le signal BOLD peut être enregistré grâce à la technologie IRM car l'hémoglobine désoxygénée est paramagnétique, c'est-à-dire qu'elle peut interagir avec un champ magnétique, alors que l'hémoglobine oxygénée ne l'est pas. Donc quand le flux sanguin augmente, la concentration en hémoglobine désoxygénée diminue, le champ magnétique devient plus homogène et le signal IRM augmente. La figure 6.1 résume comment l'activation neuronale modifie le signal IRM enregistré.

Il faut garder à l'esprit que le signal enregistré n'est pas directement l'activité neuronale mais un phénomène métabolique associé.

#### 6.1.1.2 Description des données

Les données d'IRM fonctionnelle sont de la forme $3D + t$ c'est-à-dire que ce sont des volumes d'images $3D$ de la totalité du cerveau acquis à différents instants $t$. Chaque volume est découpé en

1. au 24 juillet 2014

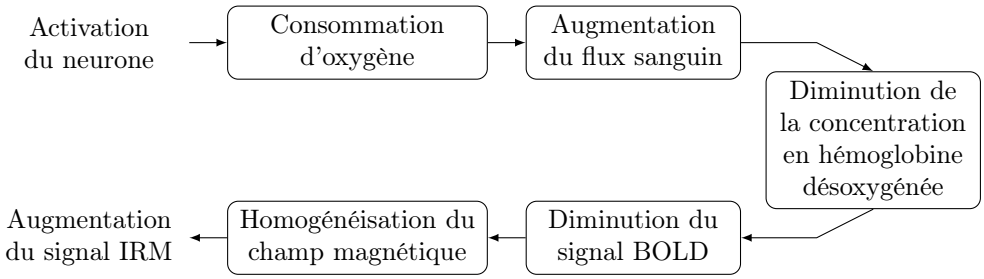



FIGURE *6.1 – Principe de l'IRM fonctionnelle : comment le signal IRM rend compte de l'activité neuronale.*

voxels (pixels $3D$). Ces voxels sont parallélépipédiques et leurs côtés sont de l'ordre du millimètre (typiquement $3mm$). La figure 6.2 donne une représentation simplifiée d'un volume de données d'IRM fonctionnelle composé de 4096 voxels.

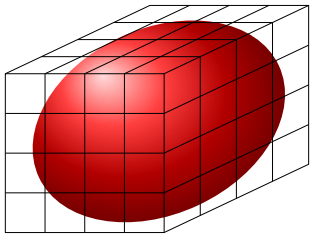


FIGURE *6.2 – Représentation simplifiée d'un volume d'IRM fonctionnelle. Le volume est constitué de 4096 voxels.*

Le temps entre deux volumes est conditionné par un des paramètres d'acquisition des données, il est de l'ordre de 2 secondes. Les données sont acquises par "tranches", c'est-à-dire sous forme d'images. La figure 6.3 montre les 40 "tranches" d'un volume de données d'IRM fonctionnelle, chaque image étant de taille $64 \times 64$ et les images ayant été acquises en partant du bas du cerveau (au niveau du cou) et en remontant jusqu'au sommet du crâne.

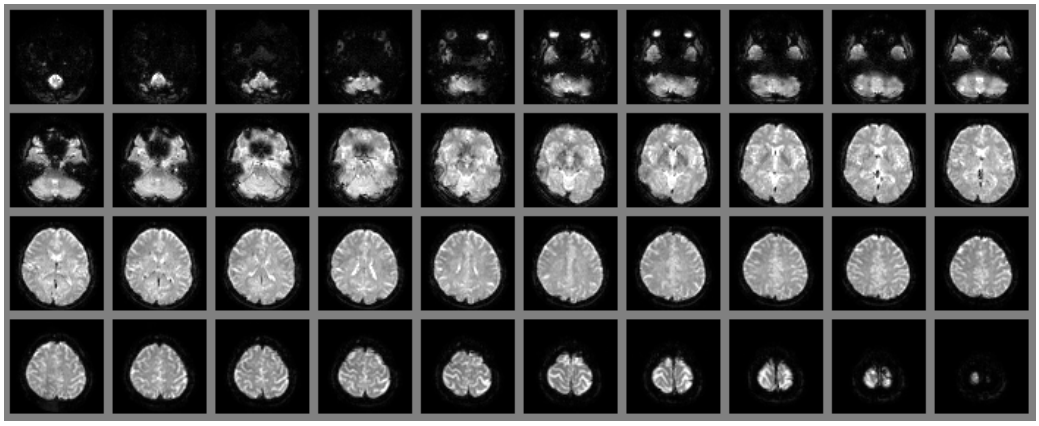


FIGURE *6.3 – Visualisation d'un volume de données d'IRM fonctionnelle par "tranches" en partant du bas du cerveau et en allant vers le haut. La visualisation est sous la forme de 40 images* $64 \times 64$.

Bien que les données soient sous la forme d'images, elles peuvent aussi être considérées comme un ensemble de signaux. A partir de l'ensemble des volumes, pour chaque voxel, un signal est extrait en prenant la valeur du voxel considéré dans chacun des volumes de données comme le montre la figure 6.4. Ce signal est échantillonné avec une période de 2 secondes (en général) et est appelé *décours temporel*.

Ce sont ces décours temporels qui sont utilisés pour étudier la connectivité fonctionnelle.

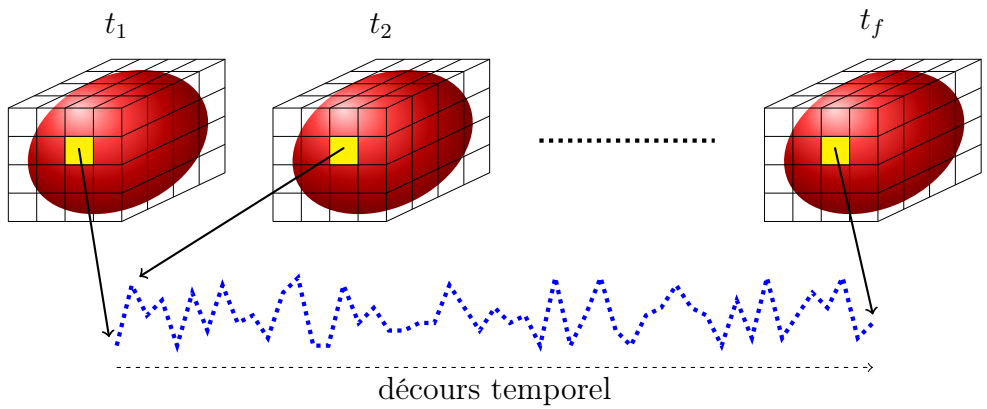



FIGURE *6.4 – Extraction d'un décours temporel de données d'IRM fonctionnelle. Le décours temporel est un signal temporel d'autant d'échantillons qu'il y a de volumes dans les données.*

### 6.1.2 Connectivité fonctionnelle

L'étude de la connectivité fonctionnelle a pour objectif de comprendre les interactions au sein du cerveau quand il entre en action. Ces interactions peuvent être étudiées à l'échelle du voxel mais cela représente une quantité très importante de données ($\simeq$ 10 000 voxels pour tout le cerveau) ou en considérant des régions cérébrales. Les régions cérébrales sont en général définies à partir d'atlas anatomiques et sont appelées ROI (regions of interest).

Pour mieux comprendre les échelles entre les éléments cérébraux impliqués dans l'étude de la connectivité fonctionnelle, la figure 6.5 illustre les ordres de grandeurs ces différents éléments.

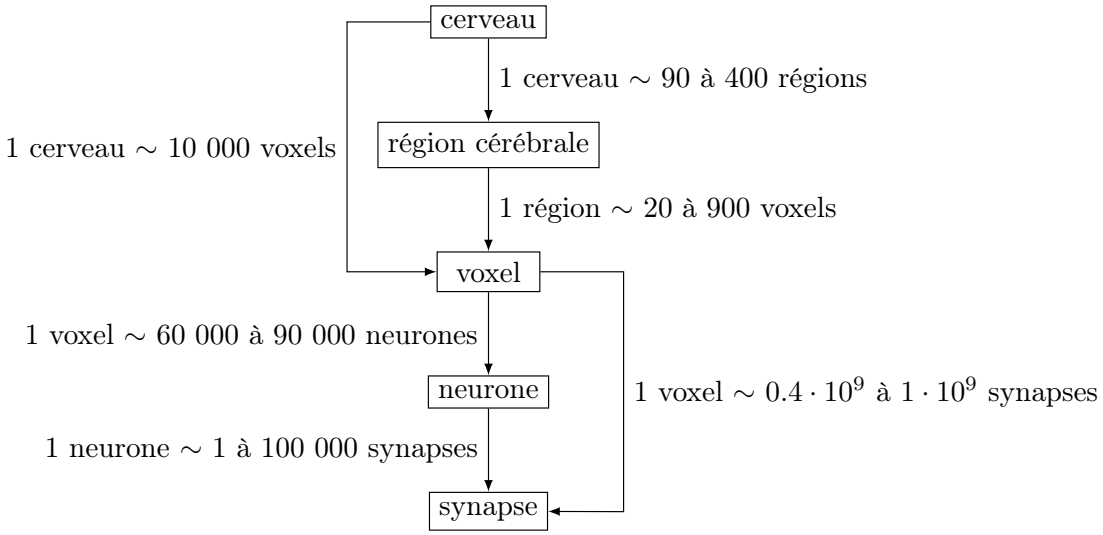



FIGURE *6.5 – Ordres de grandeurs des différents éléments cérébraux.*

Le signal observé au sein d'un voxel est une image de l'activité de plusieurs dizaines de milliers de neurones. Dans le cas de ROI, cela peut atteindre plusieurs millions de neurones. Cela illustre le fait que la connectivité fonctionnelle n'est pas étudiée à l'échelle du neurone mais à une échelle plus globale.

Il existe deux modalités d'IRM fonctionnelle :

– l'IRM fonctionnelle d'activation : le sujet étudié doit effectuer une tâche lors de l'acquisition des données puis ne rien faire puis refaire la tâche. Cette modalité permet d'identifier les

régions cérébrales impliquées lors de la réalisation de la tâche imposée.
– l'IRM fonctionnelle au repos : le sujet étudié doit ne rien faire lors de l'acquisition des données, il doit laisser son esprit vagabonder et ne pas s'endormir. Cette modalité permet d'étudier l'activité spontanée du cerveau. Cette modalité a été introduite par Biswal *et al.* [BYHH95] : Biswal a été le premier à observer chez des patients qui ne faisaient rien une activité similaire dans différentes régions associées à la motricité.

Nous nous intéressons à la modalité au repos notamment car elle permet de travailler avec des patients ne pouvant pas ou ayant des difficultés à effectuer des tâches. Il existe différentes approches pour étudier la connectivité fonctionnelle à partir de données d'IRM fonctionnelle au repos : l'approche à base de graines, l'ICA (analyse par composantes indépendantes) et les graphes.

#### 6.1.2.1 Approche à base de graines

L'approche à base de graines a pour objectif d'identifier quels voxels interagissent avec un voxel (ou une ROI) choisi par la personne réalisant l'étude. Ce voxel ou cette ROI de référence est appelée "graine". Pour identifier les régions qui interagissent avec la graine, il faut calculer la corrélation entre le décours temporel du voxel choisi (ou le décours temporel moyen des voxels dans la ROI choisie) et les signaux des autres voxels. Une fois cette corrélation calculée, un seuil est fixé pour ne conserver que les corrélations significatives : les voxels associés aux corrélations significatives sont les voxels qui interagissent avec la "graine".

Cette méthode a pour principaux inconvénients le choix de la "graine" (voxel ou ROI de référence) et le choix du seuil de la corrélation.

L'approche par graines a été utilisé par Biswal *et al.* [BYHH95] pour détecter les régions motrices ayant la même activité au repos. C'est aussi en utilisant cette approche que Raichle *et al.* [RMS+01] ont mis en évidence le réseau du mode par défaut qui est un ensemble de régions cérébrales ayant une activité similaire, qui est identifiable chez la majorité des sujets sains et qui est associé à la modalité au repos. De même, en utilisant cette méthode, Fox *et al.* [FCS+06] ont retrouvé les réseaux attentionnels dorsal et ventral déjà identifiés avec la modalité d'activation. Cohen *et al.* [CFD+08] utilisent cette méthode pour définir les régions cérébrales d'un point de vue fonctionnel et créer un atlas de régions fonctionnelles, ces derniers étant jusqu'à présent définis anatomiquement.

#### 6.1.2.2 ICA

L'ICA (Independent Component Analysis) est une méthode couramment utilisée pour la séparation de sources. Elle permet d'identifier les composantes indépendantes en maximisant l'indépendance statistique des composantes estimées.

Contrairement à l'approche à base de graines, cette méthode ne dépend pas du choix d'une région cérébrale.

McKeown *et al.* [MMB+98] furent les premiers à utiliser une méthode d'ICA sur des données d'IRM fonctionnelle. Ces données étaient acquises avec la modalité d'activation. Pour les données d'IRM fonctionnelle au repos, les pionniers sont Beckmann *et al.* [BDDS05]. Les résultats qu'ils obtiennent sont en accord avec les réseaux fonctionnels identifiés dans des études précédentes. Damoiseaux *et al.* [DRB+06] identifient, en utilisant l'ICA sur des sujets sains, les réseaux identifiés précédemment en utilisant l'approche à base de graines ou avec la modalité d'activation.

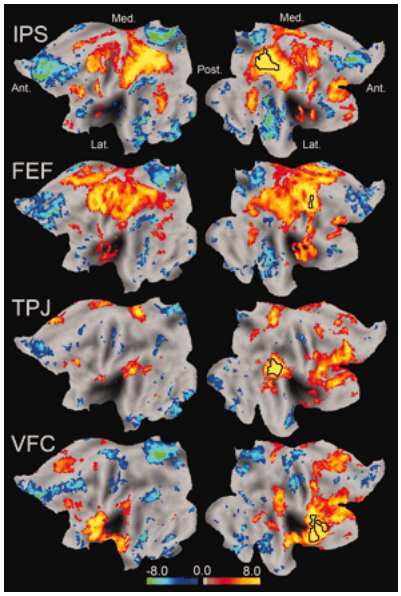



FIGURE *6.6 – Illustration issue de [FCS+06] montrant les résultats obtenus par une approche à base graines. Pour différentes graines (IPS : dans le sillon intra pariétal, FEF : dans le champ oculaire frontal, TPJ : dans la jonction temporo-pariétale et VFC : dans le cortex ventro-frontal), cette figure donne la carte des scores z des voxels significativement corrélés et anticorrélés. Les graines sont les régions dessinées en noir.*

Rosazza *et al.* [RMG+12] montrent que les résultats obtenus par l'approche à base de graines et par ICA sont similaires.

#### 6.1.2.3 Graphes

L'étude de la connectivité fonctionnelle par les graphes consiste à associer une ROI à un nœud et de tracer une arête entre les ROI ayant une forte corrélation (en valeur absolue). Le nombre d'arêtes est en général fixé et cela permet de comparer les graphes obtenus pour différents sujets en utilisant des métriques sur les graphes comme l'efficacité globale, la longueur du chemin moyen ou encore le coefficient de clustering.

Sporns *et al.* [STE00] furent les premiers à associer graphes et étude de la connectivité cérébrale. La connectivité utilisée n'était pas la connectivité fonctionnelle mais la connectivité anatomique structurelle. La première étude de connectivité fonctionnelle sur l'ensemble du cerveau humain à partir de données d'IRM fonctionnelle à l'état de repos utilisant la théorie des graphes fut menée par Achard *et al.*[ASW+06]. Cette approche utilise comme bio-marqueurs, c'est-à-dire comme métriques donnant l'information sur la pathologie d'un patient, les métriques de graphes classiques (efficacité globale, plus court chemin, coefficient de clustering,...). De nombreuses études utilisent cette approche (voir [BB11]).

Des études récentes utilisent le contexte des modèles graphiques gaussiens pour construire d'autres graphes de connectivité. Ce n'est plus la corrélation qui est utilisée pour construire les arêtes mais la corrélation partielle (cf chapitre 1). Marrelec *et al.* [MKD+06] sont les premiers à utiliser la corrélation partielle en tant que mesure des interactions entre deux régions. Ils construisent les graphes en mettant une arête entre deux régions ayant une corrélation partielle significative et se limitent à un faible nombre de régions (6). Avec l'apparition de méthodes du type *Graphical lasso* permettant d'obtenir des matrices des corrélations partielles parcimonieuses de façon rapide, même avec un grand nombre de régions, quelques études sur la totalité du cerveau

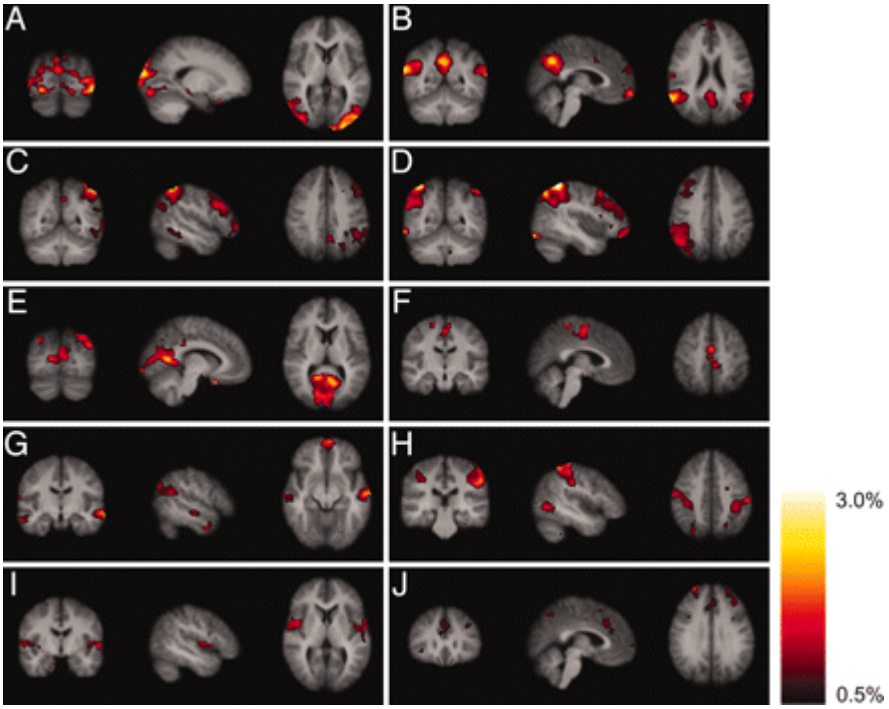



FIGURE 6.7 – *Illustration issue de [DRB+06] montrant les résultats obtenus par ICA. Cette figure représente différentes composantes indépendantes obtenues par ICA et moyennées sur 100 sujets. L'échelle est en pourcentage de changement du signal. Ces composantes peuvent être associées à des réseaux de régions cérébrales connus : par exemple la composante B correspond au réseau du mode par défaut et la composante E à une partie du réseau visuel.*

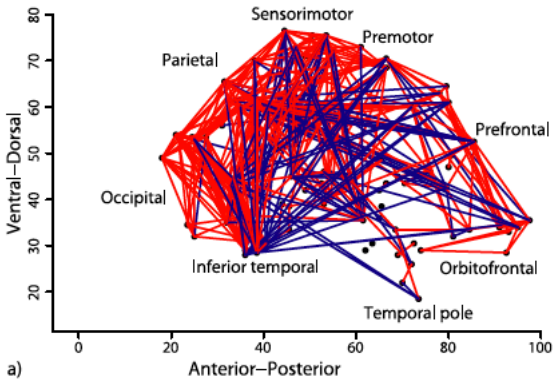



FIGURE 6.8 – *Illustration issue de [ASW+06] montrant un résultat obtenu en utilisant une approche par graphes. Cette figure représente le graphe construit en conservant les 400 corrélations les plus fortes. Les arêtes en bleu représentent des connexions entre nœuds éloignés (distance euclidienne de plus de 7.5cm), les autres étant en rouge.*

ont été menées (par exemple [RCSM12]).

#### 6.1.2.4 Discussion

Peu d'études de la connectivité fonctionnelle cérébrale sont basées sur l'étude des relations entre régions conditionnellement aux autres régions. Dans le suite, nous désignons cette approche ainsi : connectivité fonctionnelle conditionnelle. Les seules études abordant la connectivité fonc-

tionnelle conditionnelle sont les approches par graphes s'inscrivant dans le cadre des modèles graphiques gaussiens. Cependant, la majorité de ces approches utilisent le Graphical lasso qui n'est pas la méthode la plus performante comme montré dans les chapitres précédents. Si l'objectif de l'étude de la connectivité fonctionnelle est de différencier des groupes de sujets, l'approche par classification sur la divergence de Kullback-Leibler symétrisée de profils croisés normalisés présentée dans le chapitre 5 est un outil pertinent offrant une nouvelle façon d'exploiter l'information contenue dans les données d'IRM fonctionnelle.

## 6.2 Identification d'ensembles de régions cérébrales d'intérêt pertinents pour l'étude de patients dans le coma

Nous souhaitons identifier des ensembles de ROI pertinents pour l'étude de patients dans le coma. Pour cela, nous recherchons des ensembles de ROI qui permettent de différencier très nettement des patients dans le coma de sujets sains. Si ces ensembles permettent de différencier les deux groupes, cela signifie qu'en étudiant ces ensembles, nous allons étudier des ensembles de régions dont les connexions conditionnelles sont impactées par le fait que le patient est dans le coma.

### 6.2.1 Données

Dans cette partie, nous présentons les données que nous allons utiliser, c'est-à-dire les sujets scannés, les modalités d'acquisition et les pré-traitements effectués sur les données avant de les analyser. Comme nous souhaitons étudier des ensembles de ROI, nous présentons les ensembles qui ont retenu notre intérêt et pourquoi.

#### 6.2.1.1 Sujets et paramètres d'acquisition

Les données utilisées dans cette partie sont fournies par Achard *et al* [ACMV+12]. Elles se composent des données d'IRM anatomique et d'IRM fonctionnelle de patients dans le coma et de sujets sains. Le groupe des patients dans le coma comprend 17 sujets entre 21 et 82 ans et le groupe des sujets sains, 17 sujets entre 21 et 51 ans. Les données d'IRM fonctionnelle ont été acquises à l'état de repos sur une session de vingt minutes dans un appareil IRM 1.5 Tesla (Avento ; Siemens à Erlangen en Allemagne). Les données ont été acquises selon la méthode d'imagerie écho-planaire avec gradient d'écho et les paramètres suivants : temps de relaxation de 3 secondes, temps d'écho de 50 millisecondes, des voxels isotropes cubiques de 4 mm de côté, 405 volumes et 32 coupes axiales couvrant la totalité du cerveau.

#### 6.2.1.2 Pré-traitement

Les données ont été pré-traitées en utilisant le logiciel SPM8 [SPM]. Les données n'ont pas été lissées spatialement et la moyenne globale n'a pas été corrigée par régression. Les données ont subi un contrôle de qualité pour vérifier les mouvements de tête et d'autres artéfacts possibles. Les données d'IRM anatomique ont été segmentées pour obtenir la matière grise, la matière blanche et les composantes non cérébrales. Elles ont ensuite été normalisées selon le template Colin27 [HHC+98] en utilisant la méthode de recalage non-linéaire DARTEL (Diffeomorphic anatomical registration using exponential Lie algebra) [Ash07]. Le recalage fournit un champ de déformation qui est utilisé pour adapter les données d'IRM fonctionnelle à une image de segmentation du cerveau en régions basée sur l'atlas anatomique AAL (automated anatomic labeling) [TMLP+02]. Pour chaque région, le décours temporel moyen est obtenu en moyennant les décours temporels de tous les voxels de la région, pondérés par la proportion de matière grise

dans chaque voxel (proportions obtenues à partir des données d'IRM anatomique segmentées précédemment). Les mouvements de tête sont corrigés par régression sur les décours temporels des estimées des déplacements en translation et des rotations.

Comme les décours temporels des régions cérébrales sont des processus longue mémoire [BFM+04], nous appliquons une transformées en ondelette de Daubechies et nous concentrons notre analyse sur les coefficients de l'échelle 2. Cette échelle correspond à l'intervalle de fréquence 0.04-0.08 Hz. Nous choisissons cet intervalle car le domaine fréquentiel d'intérêt pour l'IRM fonctionnelle est inférieur à 0.1 Hz [BYHH95] et il contient plus de coefficients que les autres échelles respectant cette contrainte.

Tous ces pré-traitements sont les fruits de travaux antérieurs à la thèse et ne constituent pas un point qui a été remis en question.

#### 6.2.1.3 Ensembles de régions d'intérêt utilisés

Une fois les données pré-traitées, pour chaque sujet nous avons un processus de 90 variables (nombre de ROI dans l'atlas AAL) et d'une centaine d'observations (nombre de coefficients d'ondelettes à l'échelle 2).

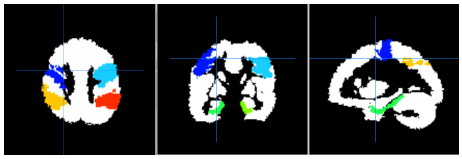
Nous sommes dans une configuration où le nombre d'observations est assimilable au nombre de variables et nous avons vu que cette configuration donne de mauvais résultats lors de l'estimation de la structure d'indépendance conditionnelle. Nous choisissons donc de nous placer dans une configuration où nous savons que les performances obtenues sont bonnes : nous réduisons le nombre de variables à 6. Nous avons choisi six ensembles de ROI selon deux critères :

- Ensembles 1-3 : choisi par expertise neurologique
- Ensembles 4-6 : régions d'intérêts où les degrés changent entre patients et sujets sains selon [ACMV+12]

Les six ensembles de ROI retenus sont les suivants :

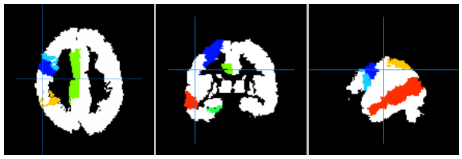
**Ensemble de ROI 1 :** Cet ensemble de ROI est constitué de 3 régions dans un hémisphère et de leurs 3 régions homotopiques dans l'autre hémisphère :

- région précentrale gauche ●
- région précentrale droite ●
- région parahippocampique gauche ●
- région parahippocampique droite ●
- région pariétale inférieure gauche ●
- région pariétale inférieure droite ●



**Ensemble de ROI 2 :** Cet ensemble de ROI est constitué de 6 régions dans le même hémisphère (hémisphère gauche) :

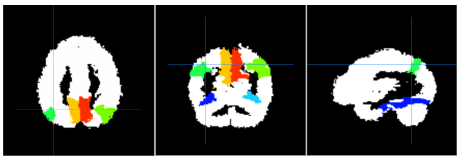
- région précentrale ●
- région parahippocampique ●
- région pariétale inférieure ●
- région frontale inférieure operculaire ●
- région du cingulaire moyen ●
- région temporale moyenne ●



**Ensemble de ROI 3 :** Cet ensemble de ROI est l'homotopique de l'ensemble 2 (c'est-à-dire les mêmes régions mais dans l'hémisphère droit).

**Ensemble de ROI 4 :** Cet ensemble de ROI est constitué de 3 régions dans un hémisphère et de leurs 3 régions homotopiques dans l'autre hémisphère :

- région fusiforme gauche
- région fusiforme droite
- région angulaire gauche
- région angulaire droite
- région du précuneus gauche
- région du précuneus droit



**Ensemble de ROI 5 :** Cet ensemble de ROI est constitué de 6 régions dans le même hémisphère (hémisphère gauche) :

- région fusiforme
- région angulaire
- région du précuneus
- région frontale inférieure orbitale
- aire motrice supérieure
- région pariétale inférieure

**Ensemble de ROI 6 :** Cet ensemble de ROI est l'homotopique de l'ensemble 5 (c'est-à-dire les mêmes régions mais dans l'hémisphère droit) .

### 6.2.2 Méthode et résultats

Nous cherchons des ensembles de ROI pertinents pour étudier les patients dans le coma. Pour que les ensembles des ROI soient pertinents, il faut qu'ils soient impactés par le fait que le patient est dans le coma. Si un ensemble est impacté, il permet de dire si les sujets sont des patients ou des sujets sains.

#### 6.2.2.1 Parmi les ensembles donnés

**Procédure**

Afin d'identifier les ensembles pertinents, pour chacun des ensembles candidats (cf. section 6.2.1.3), nous construisons pour chaque sujet un processus ayant comme variables les décours temporels associés aux ROI de l'ensemble considéré. En suite, nous classifions les différents sujets à partir de ces processus et nous analysons les performances de la classification. Pour la classification, nous utilisons la classification SVM utilisée dans le chapitre 5 avec en entrée la divergence de Kullback-Leibler symétrisée sur les profils croisés normalisés des sujets. Les profils croisés normalisés sont construits à partir des estimées des structures d'indépendance conditionnelle obtenues avec ABiGlassoMaxLoop pour les paramètres $step_\Lambda = 0.1$ et $e = 1$.

Comme les données sont composées de 17 patients et 17 sujets sains, nous choisissons d'utiliser 15 patients et 15 sujets sains pour l'ensemble d'apprentissage et d'utiliser les 4 sujets restants pour la phase de test. Nous répétons la procédure 100 fois en veillant à ne pas avoir deux fois le même ensemble d'apprentissage.

**Résultats**

La figure 6.9 présente les résultats de la classification en donnant la proportion d'itérations qui classifie correctement $x$ sujets sur les 2 sujets testés pour le groupe des patients et pour le groupe des sujets sains (contrôles).

En assimilant notre classification à un test binaire, nous pouvons poser que la condition positive est d'être un patient et que la condition négative est d'être un contrôle. Ainsi, pour mieux quantifier la qualité de la classification, nous calculons la sensibilité (performances pour le groupe des patients), la spécificité (performances pour le groupe des contrôles) et l'efficacité (performances sur la totalité des sujets) que nous reportons dans le tableau 6.1.

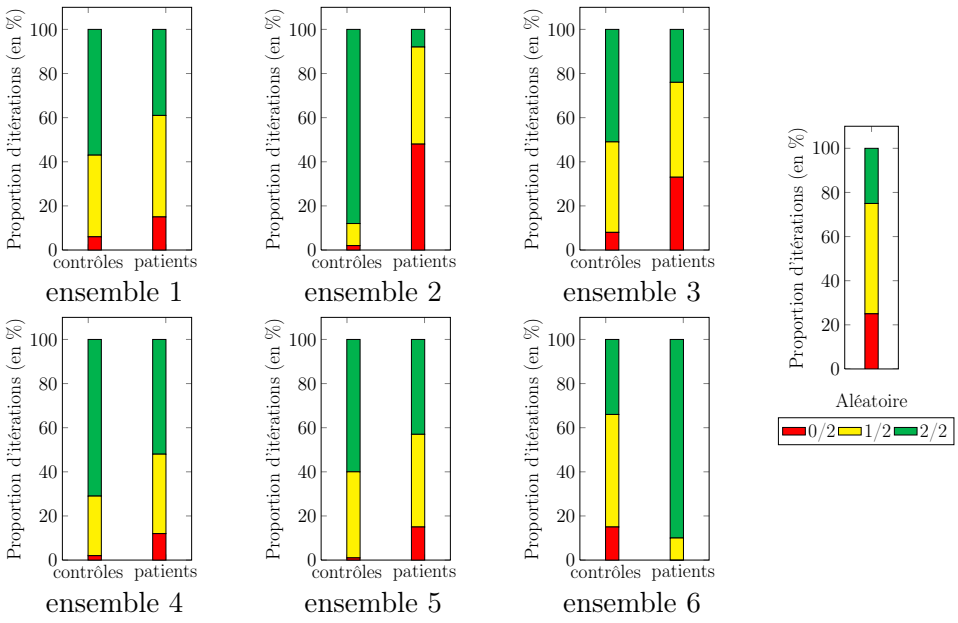



FIGURE *6.9 – Résultats de la classification. $x/2$ désigne les itérations pour lesquelles $x$ processus tests ont été bien classés sur les 2 testés.*

| ensemble de ROI | 1 | 2 | 3 | 4 | 5 | 6 | aléatoire |
|---|---|---|---|---|---|---|---|
| spécificité | 0.76 | 0.93 | 0.72 | 0.85 | 0.80 | 0.60 | 0.5 |
| sensibilité | 0.62 | 0.30 | 0.46 | 0.70 | 0.64 | 0.95 | |
| efficacité | 0.69 | 0.62 | 0.59 | 0.77 | 0.72 | 0.77 | |

TABLEAU *6.1 – Métriques pour quantifier les performances de la classification des données d'IRM fonctionnelle via SVM et en utilisant la divergence de Kullback-Leibler symétrisée sur les profils croisés normalisés. La spécificité quantifie les performances de classification des contrôles comme contrôles, la sensibilité quantifie les performances de classification des patients comme patients et l'efficacité quantifie de manière globale la bonne classification d'un sujet.*

Globalement, les six ensembles de ROI donnent des performances meilleures que si la classification avait été réalisée aléatoirement. Cependant, si nous observons les résultats obtenus plus en détails pour l'ensemble 2, nous observons que la majorité des patients sont classés comme contrôles, et pour l'ensemble 3 nous observons que les patients sont classés de manière aléatoire. Pour les autres ensembles, les résultats sont meilleurs mais en général un des deux groupes donne de meilleures performances de classification au dépend du deuxième. L'idéal serait d'avoir un ensemble de ROI pour lequel la classification donne de bons résultats quel que soit le groupe considéré.

#### 6.2.2.2 Parmi un plus grand choix d'ensembles de ROI

Existe-t-il des ensembles de 6 ROI plus pertinents que ceux étudiés dans la section précédente ? Étudier toutes les combinaisons possibles revient à étudier $\binom{90}{6} > 6 \cdot 10^8$ combinaisons. Comme cela n'est pas possible pour des raisons de temps de calcul, nous nous limitons à l'étude des ensembles dont les ROI sont dans l'hémisphère gauche et plus précisément qui font partie des quinze ROI utilisées pour générer les ensembles 2 et 5 présentés précédemment. Cela revient à étudier $\binom{15}{6} = 462$ combinaisons.

Pour chaque combinaison, nous appliquons la même procédure que pour les 6 ensembles

de ROI de la section précédente. Nous ne gardons que les résultats qui présentent à la fois de bonnes performances pour la classification des patients et pour la classification des contrôles : nous choisissons de ne garder que les ensembles de ROI pour lesquels la sensibilité et la spécificité sont supérieures à 0.9. Nous obtenons les trois ensembles suivants (rangés par ordre décroissant d'efficacité) :

A région précentrale, région fusiforme, région pariétale inférieure, région angulaire, région du précunéus et région temporale moyenne

B région précentrale, région du cingulaire moyen, région fusiforme, région pariétale inférieure, région angulaire et région temporale moyenne

C région frontale inférieure orbitale, région du cingulaire moyen, région pariétale inférieure, région angulaire, région du précunéus et région temporale moyenne

La figure 6.10 présente la proportion d'itérations qui classifie correctement $x$ sujets sur 2 pour chacun de deux groupes (patients et sujets sains). Le tableau 6.2 complète la figure en donnant la sensibilité, la spécificité et l'efficacité de la classification pour chacun des trois ensembles.

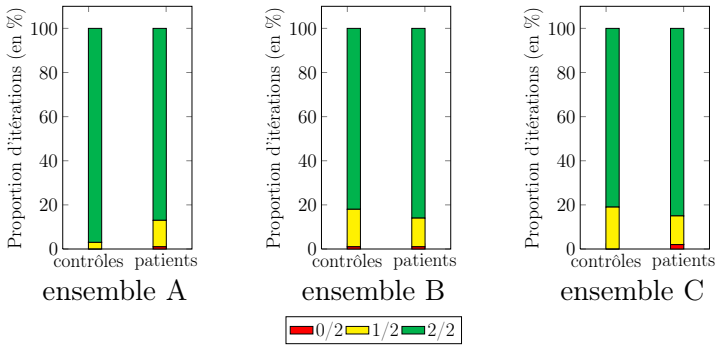



FIGURE *6.10 – Résultats de la classification. $x/2$ désigne les itérations pour lesquelles $x$ processus tests ont été bien classés sur les 2 testés.*

| ensemble | A | B | C |
|---|---|---|---|
| spécificité | 0.99 | 0.91 | 0.91 |
| sensibilité | 0.93 | 0.93 | 0.92 |
| efficacité | 0.96 | 0.92 | 0.91 |

TABLEAU *6.2 – Métriques pour quantifier les performances de la classification des données d'IRM fonctionnelle via SVM et en utilisant la divergence de Kullback-Leibler symétrisée sur les profils croisés normalisés. La spécificité quantifie les performances de classification des contrôles comme contrôles, la sensibilité quantifie les performances de classification des patients comme patients et l'efficacité quantifie de manière globale la bonne classification d'un sujet.*

Pour ces trois ensembles de ROI, la distinction entre les patients et les sujets sains est nette. Nous pouvons en déduire que ces ensembles de ROI sont pertinents pour étudier la connectivité fonctionnelle conditionnelle de patients dans le coma.

## 6.3 Discussion

Il existe peu d'études de la connectivité fonctionnelle conditionnelle cérébrale à partir de donnée d'IRM fonctionnelle, c'est-à-dire étudiant les relations entre régions cérébrales connaissant l'information contenue dans d'autres régions. Comme les approches traditionnelles s'intéressent

aux connexions directes entre deux régions, les résultats obtenus lors de ces études ne sont pas exploitables dans le cadre d'études de la connectivité fonctionnelle conditionnelle, les phénomènes observés n'étant pas les mêmes. Dans ce chapitre, notre objectif est d'identifier des ensembles de régions cérébrales d'intérêts (ROI) permettant de faire l'étude de la connectivité fonctionnelle conditionnelle sur des patients dans le coma. Afin d'évaluer la pertinence d'un ensemble de ROI, nous construisons pour chaque sujet un processus ayant comme variables les décours temporels associés aux ROI de l'ensemble considéré. En suite, nous classifions les différents sujets à partir des processus générés en utilisant l'approche développée dans le chapitre 5 et nous analysons les performances de la classification. Nous avons vu que les ensembles de ROI choisis à partir d'expertises neurologiques ou d'une étude précédente sur les mêmes données mais à partir de la corrélation (métrique de la connexion directe entre deux régions) donnent des performances mitigées, c'est-à-dire que la classification est meilleure que si elle avait été faite aléatoirement mais que la proportion de mauvaises classifications reste trop élevée (plus de 20% de mauvaises classifications). Nous avons ensuite testé toutes les combinaisons de 6 ROI possibles en choisissant les ROI parmi les 15 ROI de l'hémisphère gauche utilisées pour construire les ensembles étudiés précédemment. Parmi toutes ces combinaisons, trois ont à la fois une sensibilité et une spécificité supérieures à 0.9, c'est-à-dire que plus de 90% des patients sont classés comme patients et plus de 90 % des contrôles sont classées comme contrôles. Ces ensembles de ROI sont donc pertinents pour étudier la connectivité fonctionnelle conditionnelle dans le cadre d'étude sur des patients dans le coma.

La majorité des études de la connectivité fonctionnelle cérébrale ont lieu sur la totalité du cerveau car les chercheurs souhaitent comprendre le fonctionnement du cerveau dans sa totalité. Cependant, les données d'IRM fonctionnelle ont souvent un faible nombre d'observations. Pour obtenir plus d'observations il faudrait que le sujet reste plus longtemps dans l'IRM. Sachant que pour obtenir 400 observations, il faut que le sujet reste une vingtaine de minutes allongé dans l'IRM à ne rien faire sans dormir, augmenter la durée de l'acquisition conduirait sûrement à des données trop bruitées, le sujet pouvant se mettre à bouger ou à s'endormir. L'ordre de grandeur du nombre d'observations est donc fixé. Afin de s'affranchir des phénomènes longue mémoire présents dans les décours temporels, les données doivent subir une transformée en ondelettes qui réduit le nombre d'observations sur lesquels nous pouvons travailler à une centaine d'observations. Dans ces conditions, même en utilisant un atlas pour regrouper les voxels en régions, nous avons un problème où le nombre d'observations est similaire au nombre de variables. Si nous souhaitons estimer la structure d'indépendance conditionnelle du cerveau dans ces conditions, les résultats obtenus sont de mauvaise qualité, c'est-à-dire proche du résultat obtenu si nous avions choisi la structure aléatoirement. La question que nous nous posons est : vaut-il mieux essayer d'étudier la totalité du cerveau aux dépens de la qualité du résultat obtenu ou se focaliser sur des ensembles regroupant un faible nombre de régions mais dont nous savons que les résultats obtenus sur cet ensemble sont de bonne qualité ? Si nous souhaitons travailler sur un faible nombre de régions, il faut correctement identifier des ensembles de régions qui sont pertinents pour l'étude que nous souhaitons mener. C'est dans cette optique que nous avons développé l'approche décrite dans ce chapitre. Et nous avons montré qu'elle permet, pour l'étude de patients dans le coma, d'identifier des ensembles de régions caractéristiques des patients étudiés. Ceci ouvre des perspectives pour étudier la connectivité fonctionnelle conditionnelle dans de bonnes conditions.

# Conclusions et perspectives

L'objectif de cette thèse était de proposer une méthode pour estimer la structure d'indépendance conditionnelle d'un réseau de capteurs sous hypothèse que ce réseau de capteurs est assimilable à un processus multivarié gaussien. Le travail a donc été mené dans le cadre d'études sur les modèles graphiques gaussiens. En plus de proposer une nouvelle méthode d'estimation, nous avons proposé une approche pour évaluer les performances d'une méthode d'estimation de modèles graphiques gaussiens, nous avons montré dans quelles situations nous pouvons appliquer notre méthode, nous avons comparé ses performances à celles d'autre méthodes de la littérature et nous avons montré comment l'utiliser pour classifier des données. Ce dernier point a été appliqué pour classifier des données d'IRM fonctionnelle.

Dans le premier chapitre, nous avons présenté la notion de modèles graphiques gaussiens notamment en explicitant la notion d'indépendance conditionnelle et en introduisant des notions propres à la théorie des graphes. Dans ce même chapitre, nous avons présenté des méthodes d'estimation de modèles graphiques gaussiens afin de mettre en lumière les besoins concernant l'estimation de tels modèles. Nous sommes arrivés à la conclusion que les méthodes existantes étaient soit très rapides mais fournissaient juste un graphe solution soit elles fournissaient un graphe avec une probabilité permettant d'évaluer si ce graphe est nettement plus probable que les autres pour représenter la structure d'indépendance conditionnelle du processus étudié mais ces méthodes ont un coût de calcul pouvant être rédhibitoire. Dans le second chapitre, nous avons proposé une méthode d'évaluation des performances des méthodes d'estimation des modèles graphiques gaussiens rigoureuse afin d'évaluer la méthode que nous souhaitons développer et aussi de pouvoir la comparer à des méthodes existantes. Cette méthode comporte deux étapes : la génération de processus multivariés gaussiens dont la structure d'indépendance conditionnelle est connue et la comparaison entre la solution proposée par la méthode évaluée et la structure attendue. Pour générer les processus, nous simulons des matrices des corrélations partielles respectant une structure d'indépendance conditionnelle donnée et pour assurer que la matrice respecte bien la structure choisie, nous introduisons l'idée novatrice d'imposer que les valeurs non nulles soient supérieures en valeur absolue à un seuil défini statistiquement. Concernant la comparaison de la solution et de la structure attendue, nous utilisons deux métriques existantes, la sensibilité et la spécificité, et nous introduisons une adaptation de la distance de Hamming pour évaluer le nombre d'arêtes qui diffèrent entre les deux graphes. Dans le chapitre suivant, nous avons proposé une nouvelle méthode "ABiGlasso" dont l'objectif est de pouvoir comparer la pertinence de différents graphes à représenter la structure d'indépendance conditionnelle du processus étudié tout en limitant le nombre de graphes comparés pour limiter le temps de calcul de la méthode. Elle se base sur l'utilisation d'un score proportionnel à la probabilité du graphe de représenter la structure d'indépendance conditionnelle du processus étudié. Nous en avons proposé plusieurs variantes que nous avons analysées afin de choisir celle qui présente le meilleur compromis entre performances d'estimation et temps de calcul. Dans le chapitre 4, nous avons comparé des méthodes existantes à la variante de la méthode "ABiGlasso" retenue au chapitre précédent selon différents critères : temps de calcul, qualité de la solution estimée, aide à l'amélioration de l'estimée des matrices de corrélation et des corrélations partielles, présence ou non d'un

score permettant d'évaluer si la solution proposée est nettement plus pertinente que les autres ou si un groupe de solutions est équiprobable. Nous avons également étudié l'impact du nombre d'observations des processus étudiés principalement sur la qualité de la solution proposée, pour $p = 6$. Pour un grand nombre d'observations, les méthodes tendent à estimer parfaitement la structure attendue. Pour $n = 60$, les performances des méthodes sont meilleures que de l'estimation aléatoire mais la solution est éloignée de plusieurs arêtes de la solution attendue. Pour des processus à faible nombre de variables, notre méthode a pour avantages de fournir une solution proche de la structure attendue et également un score permettant de comparer la pertinence de plusieurs graphes à représenter la structure d'indépendance conditionnelle du processus étudié. Elle donne également les meilleurs résultats dans la configuration $p = 6$ et $n = 60$. Dans le chapitre 5, nous avons introduit une approche pour classifier en deux groupes des processus en fonction de leur structure d'indépendance conditionnelle estimée : à partir du score utilisé dans la méthode "ABiGlasso", nous construisons pour chaque processus un profil sur l'ensemble des structures estimées de tous les processus étudiés, nous comparons ces profils à l'aide de la divergence de Kullback-Leibler symétrisée et nous utilisons les valeurs de divergence obtenues en entrée d'un classifieur SVM linéaire. Les performances de cette approche ont été évaluées sur des groupes de processus ayant la même structure d'indépendance attendue intra-groupe et sur des groupes de processus ayant des structures pouvant différer d'une ou deux arêtes au sein d'un même groupe. Cette approche donne de très bons résultats même pour des processus dont la structure d'indépendance conditionnelle n'est pas parfaitement estimée. Dans le chapitre 6, nous avons appliqué l'approche développée dans le chapitre 5 à des données d'imagerie par résonance magnétique fonctionnelle pour identifier des ensembles de régions cérébrales dont la structure d'indépendance conditionnelle est impactée lorsque le patient est dans le coma. Nous parvenons à identifier des ensembles de régions pour lesquels une classification entre sujets sains et patients dans le coma est quasi-parfaite (l'efficacité est supérieure à 0.9).

## Contributions

Dans ce manuscrit nous pouvons identifier cinq contributions à l'étude de la structure d'indépendance conditionnelle de processus multivariés gaussiens. Ces contributions sont les suivantes :

1. Nous avons proposé un algorithme de simulation de processus multivariés gaussiens synthétiques qui respectent une structure d'indépendance conditionnelle donnée, c'est-à-dire dont la matrice des corrélations partielles a des valeurs nulles pour les couples d'indices correspondant à des variables indépendantes conditionnellement aux autres variables et dont les valeurs non nulles sont statistiquement significativement différentes de zéro. Cet algorithme de simulation est accompagné de métriques permettant d'évaluer les performances de méthodes d'estimation de structure d'indépendance conditionnelle en terme de nombre d'arêtes de différence entre la structure attendue et la structure estimée. Ces métriques sont la distance de Hamming, la sensibilité et la spécificité. Pour plus de détails, voir chapitre 2, section 2.1.2 pour la simulation des processus et section 2.2 pour les métriques.
2. Nous avons développé une méthode d'estimation de structure d'indépendance conditionnelle, dénommée ABiGlasso, qui permet d'estimer la structure d'indépendance conditionnelle et qui fournit un score sur les différentes structures candidates permettant ainsi d'évaluer la pertinence de la solution proposée. Cette méthode donne des résultats dans un temps raisonnable pour des processus à faibles nombres de variables. Cependant, pour des processus avec un nombre de variables supérieur à quelques dizaines, le temps de calcul est rédhibitoire. Cette méthode et ses variantes sont détaillées et évaluées sur de nombreux processus synthétiques dans le chapitre 3.
3. Nous avons mené une étude comparative des méthodes existantes (et de notre méthode)

en mettant en avant les avantages et les inconvénients de chacune des méthodes, aussi bien en terme de temps de calcul, de qualité de l'estimation ou de scores permettant d'évaluer la pertinence des solutions proposées. Cette étude a été menée sur une grand nombre de processus synthétiques. Nous avons montré que pour des processus à faible nombre de variables, notre méthode donne de très bonnes performances et a l'avantage d'informer sur la pertinence de la structure retenue. La méthode SIN présente de meilleurs résultats dans la configuration $p = 6$ et $n = 600$ mais cette configuration n'est pas représentative d'un processus réel. Cette comparaison est menée dans le chapitre 4.

4. Nous avons proposé une approche de classification de processus multivariés gaussiens basée sur l'utilisation de la structure d'indépendance conditionnelle des processus à classer et sur le score utilisé dans notre méthode "ABiGlasso". Cette approche a été appliquée à des données d'IRM fonctionnelle de patients dans le coma, permettant d'identifier des ensembles de régions cérébrales dont la structure d'indépendance conditionnelle est impactée par l'état des patients. Ces travaux sont présentés dans les chapitres 5 et 6.
5. Une Toolbox regroupant la majorité des méthodes d'estimation utilisées au cours de cette thèse a été mise en place. Cette Toolbox contient également les éléments permettant d'utiliser la procédure d'évaluation que nous avons proposée.

## Perspectives méthodologiques

### Etudes pour $p$ grand

La majorité des études menées dans ce manuscrit l'ont été sur des processus de six variables. Une des perspectives possibles est d'étendre les résultats obtenus pour six variables à un nombre plus important de variables. A la fin du chapitre 4, nous avons présenté les résultats obtenus pour un processus avec $p = 100$ variables. Nous avons vu que certaines méthodes ne pouvaient pas être appliquées à cause de leur temps de calcul. Nous avons également vu que la méthode "ABiGlasso" que nous avons proposée n'est plus pertinente pour des grandes valeurs de $p$.

Bien que la méthode "ABiGlasso" ne soit pas applicable pour de grandes valeurs de $p$, la méthode dite de *comparaison de processus à partir de leur structure d'indépendance conditionnelle* peut quand même être utilisée pour comparer des processus ayant un grand nombre de variables : il suffit d'estimer les structures d'indépendance conditionnelle à partir d'une autre méthode plus rapide et plus performante pour des grandes valeurs de $p$ puis d'utiliser le score que nous avons introduit pour la méthode "ABiGlasso". Supposons que nous souhaitions comparer 40 processus (ce qui est courant comme ordre de grandeur dans le cas d'une étude à partir de données d'IRM fonctionnelle), nous avons au maximum $40 \times 40$ scores à calculer. Ce maximum est atteint quand aucun des processus n'a la même structure qu'un autre. Pour des processus à $p = 100$ variables, cela prend environ 11 heures de calculer tous les scores. Une fois ces scores obtenus, la procédure est la même que celle développée dans le chapitre 5 : nous construisons des profils sur l'ensemble des structures estimées, nous comparons ces profils croisés normalisés avec la divergence de Kullback-Leibler symétrisée et nous faisons la classification à partir de cette divergence. Notons que le temps de calcul des scores peut être réduit en parallélisant ce calcul.

Pour travailler avec des processus ayant un grand nombre de variables, une alternative à l'étude du processus dans son ensemble serait de hiérarchiser l'estimation de sa structure d'indépendance conditionnelle : étudier indépendamment plusieurs sous-ensembles de variables du processus puis les regrouper pour obtenir la structure globale du processus. Pour utiliser une telle approche, il faut étudier comment séparer les variables du processus et comment reconstruire la structure globale à partir des sous-structures obtenues. La notion de graphe décomposable semble être indispensable pour appréhender cet aspect de l'étude.

### Autres

Une autre perspective est d'étudier si il est plus intéressant d'estimer la structure d'indépendance conditionnelle directement à partir de la totalité du processus ou en faisant du bootstrap sur le nombre d'observations. Par exemple, pour un processus à 600 observations, est-il plus intéressant d'estimer le structure d'indépendance conditionnelle directement ou en faisant du bootstrap sur des portions du processus de 60 observations ?

## Perspectives applicatives

### En IRM fonctionnelle

L'étude menée dans le chapitre 6 est uniquement exploratoire. Elle met en lumière l'existence d'ensembles de régions cérébrales qui permettent de distinguer des patients dans le coma de sujets sains. Cette étude nécessite d'être approfondie notamment par une expertise neurologique pour savoir pourquoi et comment ces ensembles de régions sont impactés par le fait que les patients sont dans le coma.

L'étude de la connectivité fonctionnelle conditionnelle, c'est-à-dire l'étude de la connectivité fonctionnelle en utilisant une approche basée sur l'étude de la dépendance conditionnelle, n'est pas couramment utilisée à notre connaissance et mérite d'être plus utilisée. En effet, nous nous sommes intéressés à l'étude des structures d'indépendance conditionnelle afin de compléter les informations apportées par l'étude de l'indépendance statistique classique. L'approche du chapitre 6 peut aussi être appliquée à d'autres études neurologiques utilisant l'IRM fonctionnelle afin de compléter les études existantes puisqu'elle identifie des ensembles de régions cérébrales pertinentes pour l'étude de la connectivité fonctionnelle conditionnelle.

### En général

L'approche de classification présentée dans le chapitre 5 peut être utilisée pour faire de la classification comme nous l'avons fait sur les données d'IRM fonctionnelle dans le chapitre 6. Cette approche souffre moins du passage aux grandes dimensions : si nous identifions un sous-groupe de variables dont la structure d'indépendance conditionnelle permet de différencier deux groupes de processus, même si ces processus ont un nombre important de variables, nous pouvons faire de la classification avec la procédure présentée. Cette procédure peut s'appliquer à n'importe quel type de réseaux de capteurs.

## Publications

**Conférence nationale :** A. Costard, S. Achard, O.J.J. Michel, P. Borgnat, and P. Abry. Estimation bayésienne asymptotique de la structure d'un graphe initialisée par Graphical lasso. In *Actes du colloque GRETSI*, 2013

**Conférence internationale :** A. Costard, S. Achard, O.J.J. Michel, P. Borgnat, and P. Abry. Data comparison using Gaussian Graphical Models. In *IEEE International Conference in Signal Processing (ICSP)*, 2014

**Article de journal :** en cours de rédaction au moment de l'édition du présent manuscrit.

# Bibliographie


[ACMV+12] S. Achard, C.Delon-Martin, P.E. Vértes, F. Renard, M. Schenck, F. Schneider, C. Heinrich, S. Kremer, and E.T. Bullmore. Hubs of brain functional networks are radically reorganized in comatose patients. *Proceedings of the National Academy of Sciences*, 109 :20608–13, 2012.

[ACWK09] H. Armstrong, C.K. Carter, K.F.K. Wong, and R. Kohn. Bayesian covariance matrix estimation using a mixture of decomposable graphical models. *Statistics and Computing*, 19(3) :303–316, 2009.

[AG09] T. Alun and P.J. Green. Enumerating the junction trees of a decomposable graphs. *Journal of Computational and Graphical Statistics*, 18 :930 – 940, 2009.

[AKM05] A. Atay-Kayis and H. Massam. A Monte Carlo method to compute the marginal likelihood in non decomposable graphical Gaussian models. *Biometrika*, 92 :317–335, 2005.

[And03] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, 2003.

[Ash07] J. Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1) :95–113, 2007.

[ASW+06] S. Achard, R. Salvador, B. Whitcher, J. Suckling, and E. Bullmore. A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *The Journal of Neuroscience*, 26 :63–72, 2006.

[AT05] V. Arvind and J. Toran. Isomorphism testing : perspective and open problems. *Bulletin of the European Association for Theoretical Computer Science*, pages 66–84, 2005.

[BB11] E.T. Bullmore and D.S. Bassett. Brain graphs : Graphical models of the human brain connectome. *Annual Review of Clinical Psychology*, 7 :113–140, 2011.

[BDDS05] C.F. Beckmann, M. DeLuca, J.T. Devlin, and S.M. Smith. Investigations into resting-state connectivity using Independent Component Analysis. *Philosophical transactions of the royal society*, 360 :1001–1013, 2005.

[BFM+04] E. Bullmore, J. Fadili, V. Maxim, L. Snedur, B. Whitcher, J. Suckling, M. Brammer, and M. Breakspear. Wavelets and functional magnetic resonance imaging of the human brain. *NeuroImage*, 23 :S234–S249, 2004.

[BGd08] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9 :485–516, 2008.

[BGH09] Y. Baraud, C. Giraud, and S. Huet. Gaussian model selection with an unknown variance. *The Annals of Statistics*, 37 :630–672, 2009.

[BPC+10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3 :1–122, 2010.

[BYHH95] B. Biswal, F. Zerrin Yetkin, V.M. Haughton, and J.S. Hyde. Functional connectivity in the motor cortex of resting human brain using echo-planar MRI. *Magnetic Resonance in Medicine*, 34(4) :537–541, 1995.

[CAM+13] A. Costard, S. Achard, O.J.J. Michel, P. Borgnat, and P. Abry. Estimation bayésienne asymptotique de la structure d'un graphe initialisée par Graphical lasso. In *Actes du colloque GRETSI*, 2013.

[CAM+14] A. Costard, S. Achard, O.J.J. Michel, P. Borgnat, and P. Abry. Data comparison using Gaussian Graphical Models. In *IEEE International Conference in Signal Processing (ICSP)*, 2014.

[CFD+08] A.L. Cohen, D.A. Fair, N.U.F. Dosenbach, F.M. Miezin, D. Dierker, D.C. Van Essen, B.L. Schlaggar, and S.E. Petersen. Defining functional areas in individual human brains using resting functional connectivity MRI. *NeuroImage*, 41 :45 – 57, 2008.

[CPJ10] C.M. Carvalho, N.G. Polson, and J.Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97 :465–480, 2010.

[CR06] 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.

[CR09] R. Castelo and A. Roverato. Reverse engineering molecular regulatory networks from microarray data with qp-graphs. *Journal of Computational Biology*, 16(2) :213–217, 2009.

[CS09] C.M. Carvalho and J.G. Scott. Objective Bayesian model selection in Gaussian graphical models. *Biometrika*, pages 1–16, 2009.

[CST00] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[Dem72] A.P. Dempster. Covariance selection. *Biometrics*, 28 :157–175, 1972.

[DGR03] P. Dellaportas, P. Giudici, and G. Roberts. Bayesian inferance for nondecomposable graphical Gaussian models. *Sankhya : The Indian Journal of Statistics*, 65 :43–55, 2003.

[DL93] P. Dawid and S.L. Lauritzen. Hyper Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics*, 21 :1272–1317, 1993.

[DM12] S. Donnet and J.-M. Marin. An empirical Bayes procedure for the selection of Gaussian graphical models. *Statistics and Computing*, 22 :1113–1123, 2012.

[DP04] M. Drton and M.D. Perlman. Model selection for Gaussian concentration graphs. *Biometrika*, 91 :591–602, 2004.

[DP07] M. Drton and M.D. Perlman. Multiple testing and error control in Gaussian Graphical Model selection. *Statistical Science*, 22 :430–449, 2007.

[DP08] M. Drton and M.D. Perlman. A SINful approach to Gaussian graphical model selection. *Journal of Stastical planning and inference*, 138 :1179–1200, 2008.

[DRB+06] J.S Damoiseaux, S.A.R.B Rombouts, F. Barkhof, P. Scheltens, C.J. Stam, S.M. Smith, and C.F. Beckmann. Consistent resting-state networks across healthy subjects. *Proceedings of the National Academy of Sciences*, 103 :13848–13853, 2006.

[DT08] A. Dalalyan and A.B. Tsybakov. Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity. *Machine Learning*, 72 :39–61, 2008.

[Edw00] D. Edwards. *Introduction to Graphical Modelling.* Springer-Verlag, 2000.

[FCS+06] M.D. Fox, M. Corbetta, A.Z. Snyder, J.L. Vincent, and M.E. Raichle. Spontaneous neuronal activity distinguishes human dorsal and ventral attention systems. *Proceedings of the National Academy of Sciences*, 103 :10046–10051, 2006.

[FFW09] J. Fan, Y. Feng, and Y. Wu. Network exploration via the adaptive lasso and SCAD penalties. *The Annals of Applied Statistics*, 3(2) :521–541, 2009.

[FHT08] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the Graphical lasso. *Biostatistics*, 9 :432–441, 2008.

[FJ12] A.M. Fitch and B. Jones. The cost of using decomposable Gaussian graphical models for computational convenience. *Computational Statistics and Data Analysis*, 56 :2430–2441, 2012.

[GG99] P. Giudici and P.J. Green. Decomposable graphical Gaussian model determination. *Biometrika*, 86 :785–801, 1999.

[GHV12] C. Giraud, S. Huet, and N. Verzelen. Graph selection with GGMselect. *Statistical Applications in Genetics and Molecular Biology*, 11 :1544–6115, 2012.

[Gir08] C. Giraud. Estimation of Gaussian graphs by model selection. *Electronic Journal of Statistics*, 2 :542–563, 2008.

[Giu96] P. Giudici. Learning in graphical Gaussian models. *Bayesian Statistics*, 5 :621–628, 1996.

[GS13] P. Giudici and A. Spelta. Graphical network models for international financial flows. DEM Working Papers Series 052, University of Pavia, Department of Economics and Management, 2013.

[HB11] A. Hero and B.Rajaratnam. Large scale correlation screening. *Journal of the American Statistical Association*, 106 :1540–1552, 2011.

[HHC+98] C.J. Holmes, R. Hoge, L. Collins, R. Woods, A.W. Toga, and A.C. Evans. Enhancement of MR images using registration for signal averaging. *Journal of Computer Assisted Tomography*, 22(2) :324–333, 1998.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction.* Springer, 2009.

[JCD+05] B. Jones, C. Carvalho, A. Dobra, C. Hans, C. Carter, and M. West. Experiments in stochastic computation for high-dimensional graphical models. *Statistical Science*, 20 :388 – 400, 2005.

[KL04] E. Kuhn and M. Lavielle. Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM : PS*, 8 :115–131, 2004.

[Lau96] S.L. Lauritzen. *Graphical models*. Oxford University press, 1996.

[LG06] H. Li and J. Gui. Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7 :302–317, 2006.

[LM01] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87 :198701, 2001.

[MB06a] G. Marrelec and H. Benali. Asymptotic Bayesian structure learning using graph supports for Gaussian Graphical Models. *Journal of Multivariate Analysis*, 97 :1451–1466, 2006.

[MB06b] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34 :1435–1462, 2006.

[Mei08] N. Meinshausen. A note on the lasso for Gaussian Graphical Model selection. *Statistics and Probability letters*, 78 :880 – 884, 2008.

[MKD+06] G. Marrelec, A. Krainik, H. Duffau, M. Pélégrini-Issac, S. Lehéricy, J. Doyon, and H. Benali. Partial correlation for functional brain interactivity investigation in functional MRI. *NeuroImage*, 32 :228 – 237, 2006.

[MMB+98] M.J. McKeown, S. Makeig, G.G. Brown, T.-P. Jung, S.S. Kindermann, A.J. Bell, and T.J. Sejnowski. Analysis of fMRI data by blind separation into Independent Spatial Components. *Human Brain Mapping*, 6 :160–188, 1998.

[Mou94] P. Moulin. Wavelet thresholding techniques for power spectrum estimation. *Signal Processing, IEEE Transactions on*, 42 :3126–3136, 1994.

[RCSM12] S. Ryali, T. Chen, K. Supekar, and V. Menon. Estimation of functional connectivity in fMRI data using stability selection-based sparse partial correlation with elastic net penalty. *NeuroImage*, 59 :3852–3861, 2012.

[RMG+12] C. Rosazza, L. Minati, F. Ghielmetti, M.L. Mandelli, and M.G. Bruzzone. Functional connectivity during resting-state functional MR imaging : Study of the correspondence between Independent Component Analysis and Region-of-Interest based methods. *American Journal of NeuroRadiology*, 33 :180–187, 2012.

[RMS+01] M.E. Raichle, A.M. MacLeod, A.Z. Snyder, W.J., D.A. Gusnard, and G.L. Shulman. A default mode of brain function. *Proceedings of the National Academy of Sciences*, 98 :676–682, 2001.

[Rov99] A. Roverato. *Asymptotic prior to posterior analysis for graphical Gaussian models*, pages 335–342. Springer-Verlag, 1999.

[Rov02] A. Roverato. Hyper inverse Wishart distribution for non-decomposable graphs and its application to Bayesian inference for Gaussian Graphical Models. *Scandinavian Journal of Statistics*, 29(3) :391–411, 2002.

[RW98] A. Roverato and J. Whittaker. The Isserlis matrix and its application to non-decomposable graphical Gaussian models. *Biometrika*, 85(3) :711–725, 1998.

[RWRY11] P. Ravikumar, M.J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5 :935–980, 2011.

[SC08] J.G. Scott and C.M. Carvalho. Feature-inclusion stochastic search for Gaussian Graphical Models. *Journal of Computational and Graphical Statistics*, 17 :790–808, 2008.

[Sid67] Z. Sidak. Rectangular confidence regions for the means of multivariate normal distributions. *The Annals of Mathematical Statistics*, 62 :626–633, 1967.

[SPM] Spm.

[STE00] O. Sporns, G. Tononi, and G.M. Edelman. Theoretical neuroanatomy : Relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cerebral Cortex*, 10 :127–141, 2000.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58 :267–288, 1996.

[Tib11] R. Tibshirani. Regression shrinkage and selection via the lasso : a retrospective. *Journal of the Royal Statistical Society*, 73 :273–282, 2011.

[TMLP+02] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage*, 15(1) :273–289, 2002.

[TS14] R. Talluri and S.Shete. Gaussian Graphical Models for phenotypes using pedigree data and exploratory analysis using networks with genetic and nongenetic factors based on genetic analysis workshop 18 data. In *BMC Proceedings*, 2014.

[Wan12] H. Wang. Bayesian Graphical lasso models and efficient posterior computation. *Bayesian Analysis*, 7 :771–790, 2012.

[WB06] A. Wille and P. Bühlmann. Low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, 5, 2006.

[Wer76] N. Wermuth. Analogies between multiplicative models in contingency tables and covariance selection. *Biometrics*, 32 :95–108, 1976.

[Whi90] J. Whittaker. *Graphical models in applied multivariate statistics*. Wiley, 1990.

[Wis28] J. Wishart. The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, 20A :32–52, 1928.

[WS98] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393 :440–442, 1998.

[Zel86] A. Zellner. On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *Bayesian inference and decision techniques : Essays in Honor of Bruno De Finetti*, 6 :233–243, 1986.

[ZFLH14] T. Zerenner, P. Friederichs, K. Lehnertz, and A. Hense. A Gaussian Graphical Model approach to climate networks. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 24 :023103, 2014.

# Annexe A

# Démonstration de $\beta_{ij} = 0 \Leftrightarrow \text{cov}(X_i, X_j | X_{V\setminus\{i,j\}}) = 0$

**Théorème A.0.1.** *Soit $T = (X, Y, Z)$ avec $T \sim \mathcal{N}_{p+m+l}(0, \Sigma)$ et $n$ observations. $X$ est de taille $p \times n$, $Y$ de taille $m$ et $Z$ de taille $l$. La régression linéaire de $Y$ par rapport à $Z$ et $X$ s'écrit de la façon suivante :*

$$Y = \beta_Z Z + \beta_X X + \epsilon_Y.$$

*Si $X$ et $Y$ sont indépendants conditionnellement à $Z$ alors $\beta_X = 0$.*

*Démonstration.*

$$\begin{aligned}\widehat{Y}(Z, X) &= \text{cov}\left(Y, \begin{bmatrix} Z \\ X \end{bmatrix}\right) \text{var}(Z, X)^{-1} \begin{bmatrix} Z \\ X \end{bmatrix} \\ &= [\text{cov}(Y, Z) \quad \text{cov}(Y, X)] \begin{bmatrix} \text{var}(Z) & \text{cov}(Z, X) \\ \text{cov}(X, Z) & \text{var}(X) \end{bmatrix}^{-1} \begin{bmatrix} Z \\ X \end{bmatrix}\end{aligned}$$

où :
- $\text{cov}(Y, Z) = \mathbb{E}[YZ^t]$ est une matrice $m \times l$
- $\text{var}(Z) = \mathbb{E}[ZZ^t]$ est une matrice $l \times l$

D'après le lemme de l'inverse de la variance (Whittaker, 5.7)

$$\begin{bmatrix} \text{var}(Z) & \text{cov}(Z, X) \\ \text{cov}(X, Z) & \text{var}(X) \end{bmatrix}^{-1} = \begin{bmatrix} \text{var}(Z)^{-1} + B^t \text{var}(X|Z)^{-1} B & -B^t \text{var}(X|Z)^{-1} \\ -\text{var}(X|Z)^{-1} B & \text{var}(X|Z)^{-1} \end{bmatrix}$$

où $\text{var}(X|Z) = \mathbb{E}\left[[X - \widehat{X}(Z)][X - \widehat{X}(Z)]^t\right]$ et $B = \text{cov}(X, Z)\text{var}(Z)^{-1}$.

Le coefficient matriciel de $X$ dans la régression $\widehat{Y}(Z, X)$ est donc :

$$\begin{aligned}\beta_X &= -\text{cov}(Y, Z) B^t \text{var}(X|Z)^{-1} + \text{cov}(Y, X)\text{var}(X|Z) \\ &= \left(-\text{cov}(Y, Z) B^t + \text{cov}(Y, X)\right) \text{var}(X|Z)^{-1} \\ &= \left(-\text{cov}(Y, Z)\text{var}(Z)^{-t}\text{cov}(X, Z)^t + \text{cov}(Y, X)\right) \text{var}(X|Z)^{-1}\end{aligned}$$

.

Comme :

- $\mathrm{var}(Z)^{-t} = \mathrm{var}(Z)^{-1}$
- $\mathrm{cov}(X, Z)^t = \mathrm{cov}(Z, X)$

$$\beta_X = \big(\mathrm{cov}(Y, X) - \mathrm{cov}(Y, Z)\mathrm{var}(Z)^{-1}\mathrm{cov}(Z, X)+\big)\,\mathrm{var}(X|Z)^{-1}$$

D'après (Whittaker 5.5) :

$$\beta_X = \mathrm{cov}(Y, X|Z)\mathrm{var}(X|Z)^{-1}$$

**Conclusion :** $\beta_X = 0 \Leftrightarrow \mathrm{cov}(Y, X|Z) = 0$

□

# Annexe B

# Détails de la méthode par tests-multiples SIN

La méthode SIN proposée par Drton et Perlman [DP04] est un test statistique sur les valeurs de la matrice des corrélations partielles. Cette méthode teste si les valeurs de la matrice des corrélations partielles sont significativement non nulles en utilisant une approche par tests multiples. Cette méthode est détaillée dans la présente annexe. La méthode est présentée dans le chapitre 1, à la section 1.2.1.

Leur approche est basée sur la transformée en $z$ de Fisher :

$$z : \begin{cases} [-1,1] & \rightarrow & \mathbb{R} \\ r & \rightarrow & \frac{1}{2}\ln(\frac{1+r}{1-r}) \end{cases} \tag{B.1}$$

et l'inégalité de Sidak [Sid67] :

$$\mathbb{P}(x_1 \leq sc_1, \ldots, x_k \leq sc_k) \geq \prod_{i=1}^{k} \mathbb{P}(x_i \leq sc_i) \tag{B.2}$$

Selon [And03], sections 4.2.3 et 4.3.3

$$\sqrt{n_p}\big(z(\pi_{ij}^{emp}) - z(\pi_{ij})\big) \overset{a}{\sim} \mathcal{N}(0,1) \tag{B.3}$$

avec $n_p = n - 3 - (p-2)$ où $n$ est le nombre d'observations, $p$ le nombre de variables et $\pi_{ij}^{emp}$ la valeur de la corrélation partielle empirique entre $X_i$ et $X_j$. Notons que $\pi_{ij} = 0 \Leftrightarrow z(\pi_{ij}) = 0$.

Nous voulons contrôler la probabilité $\mathbb{P}(\pi_{ij} = 0)$. Cela équivaut à contrôler la probabilité $\mathbb{P}\Big( -\sqrt{n_p}|z(\pi_{ij}^{emp})| \leq \sqrt{n_p}\big(z(\pi_{ij}^{emp}) - z(\pi_{ij})\big) \leq \sqrt{n_p}|z(\pi_{ij}^{emp})|\Big)$.

Soit $X$ une variable suivant la loi normale $\mathcal{N}(0,1)$, $\Phi_{0,1}$ correspond à la fonction de répartition associée à la loi normale :

$$\Phi_{0,1}(x) = \frac{1}{2\pi}\int_{-\infty}^{x} \exp(-\frac{1}{2}t^2)dt \tag{B.4}$$

Sachant que $\Phi_{0,1}(-x) = 1 - \Phi_{0,1}(x)$ :

$$\mathbb{P}(-x \leq X \leq x) = \mathbb{P}(X \leq x) - \mathbb{P}(X \leq -x) \quad = \Phi_{0,1}(x) - \Phi_{0,1}(-x) = 2\Phi_{0,1}(x) - 1 \tag{B.5}$$

D'après (B.5) :

$$\mathbb{P}\Big( -\sqrt{n_p}|z(\pi_{ij}^{emp})| \leq \sqrt{n_p}\big(z(\pi_{ij}^{emp}) - z(\pi_{ij})\big) \leq \sqrt{n_p}|z(\pi_{ij}^{emp})|\Big) = 2\Phi_{0,1}\big(\sqrt{n_p}|z(\pi_{ij}^{emp})|\big) - 1.$$

D'après l'inégalité de Sidak :

$$\begin{aligned}
&\mathbb{P}\Big(\sqrt{n_p}|z(\pi_{ij}^{emp}) - z(\pi_{ij})| \leq \sqrt{n_p}\zeta(\alpha), 1 \leq i < j \leq p\Big) \\
&\qquad \geq \prod_{1\leq i<j\leq p} \mathbb{P}\Big(\sqrt{n_p}|z(\pi_{ij}^{emp}) - z(\pi_{ij})| \leq \sqrt{n_p}\zeta(\alpha)\Big) \\
&\qquad = \Big(2\Phi_{0,1}\Big(\sqrt{n_p}\zeta(\alpha)\Big) - 1\Big)^{\frac{p(p-1)}{2}} \\
&\qquad = 1 - \alpha
\end{aligned}$$

L'hypothèse $\mathcal{H}_0$ est donc rejetée pour chaque couple $(i, j)$ si $\Big(2\Phi_{0,1}\big(\sqrt{n_p}|z(\pi_{ij}^{emp})|\big) - 1\Big)^{\frac{p(p-1)}{2}}$ est supérieure à $1 - \alpha$. Ils appellent $1 - \Big(2\Phi_{0,1}(\sqrt{n_p}|z(\pi_{ij}^{emp})|\big) - 1\Big)^{\frac{p(p-1)}{2}}$ la $p$-valeur simultanée de $z(\pi_{ij}^{emp})$.

Le graphe $G(\alpha)$ est construit de la façon suivante, pour $1 \leq i < j \leq p$ :

$$\begin{cases} (i,j) \notin E & \text{si } 1 - \Big(2\Phi_{0,1}(\sqrt{n_p}|z(\pi_{ij}^{emp})|\big) - 1\Big)^{\frac{p(p-1)}{2}} \geq \alpha \\ (i,j) \in E & \text{si } 1 - \Big(2\Phi_{0,1}(\sqrt{n_p}|z(\pi_{ij}^{emp})|\big) - 1\Big)^{\frac{p(p-1)}{2}} < \alpha \end{cases}$$

Ils prouvent que :

$$\liminf_{n\to\infty} \mathbb{P}(G(\alpha) = G) \geq 1 - \alpha. \tag{B.6}$$

Dans [DP07], ils introduisent une amélioration en utilisant la procédure de Holm qui prend en compte l'ordonnancement des $p-$valeurs simultanées. Le graphe construit à partir de cette amélioration respecte également (B.6).

# Annexe C

# Détails de la méthode Bayesian Adaptive Glasso

Dans cette annexe, nous présentons le modèle utilisé par Wang [Wan12] pour estimer la matrice de précision à partir du Graphical lasso et en utilisant une approche bayésienne. Cette méthode est introduite dans le chapitre 1, section 1.2.2.2.

L'estimateur Graphical lasso est équivalent au maximum a posteriori du modèle suivant :

$$\begin{cases} \mathbb{P}(\mathbf{X}|K=\Sigma^{-1}) = \varphi_{\mathbf{0},\Sigma}(\mathbf{X}) \\ \mathbb{P}(K|\rho) = cst^{-1} \prod_{i<j}\Big(\text{dexp}(k_{ij}|\rho)\Big) \prod_{i=1}^{p}\Big(\exp(k_{ii}|\frac{\rho}{2})\Big) \mathbb{1}_{K\in\mathcal{M}^+} \end{cases} \tag{C.1}$$

où $\varphi_{\mu,\Sigma}(\mathbf{X})$ est la loi gaussienne de moyenne $\mu$, de covariance $\Sigma$ appliquée au processus $\mathbf{X}$ et dexp correspond à la fonction de densité d'une double exponentielle de la forme : $\mathbb{P}(x) = \frac{\rho}{2}\exp(-\rho|x|)$ et $\rho = \lambda/n$. L'indicatrice $\mathbb{1}_{K\in\mathcal{M}^+}$ impose que la solution soit une matrice définie positive.

La première équation du modèle C.1 correspond au fait que notre processus est multivarié gaussien, la deuxième équation correspond à l'a priori bayésien d'une pénalisation lasso, c'est-à-dire un a priori de Laplace. Ce dernier a priori est introduit par Tibshirani [Tib96] en même temps que la méthode *lasso*.

Le résultat de cette approche est une estimée de la matrice de précision, pas nécessairement parcimonieuse et une estimée de la valeur du paramètre de pénalisation $\lambda$ pour chacun des éléments de la matrice de précision.

# Annexe D

# Détails de la méthode GGMselect

La méthode de Giraud *et al.* [GHV12], introduite dans le chapitre 1, section 1.2.3, utilise un critère pour comparer des graphes et limite les graphes à comparer à une famillle de graphes. Le critère est présenté en détail dans cette annexe, comme les différentes familles proposées par les auteurs.

Nous avons vu dans la section 1.1.1.3 du chapitre 1 que pour un processus multivarié gaussien $\mathbf{X}$ l'estimateur linéaire de $X_i$ sachant les autres variables vaut $\widehat{X}_i(X_{V\setminus\{i\}}) = \sum_{\substack{k=1\\k\neq i}}^{p} \beta_{ik} X_k$ et dire que deux variables $X_i$ et $X_j$ sont indépendantes conditionnellement aux autres variables du processus équivaut à dire que le coefficient de régression $\beta_{ij}$ est nul. Dans la suite nous notons $B$ la matrice $p \times p$ des coefficients de régressions linéaires $[\beta_{ij}]$ avec $\beta_{ii} = 0$.

## D.1 Critère pour la sélection de graphe

Giraud *et al.* [GHV12] considèrent une famille de graphes $\widehat{\mathcal{G}}$. Nous rappelons que le processus $\mathbf{X}$ est assimilable à une matrice $n \times p$ et que $X_i$ désigne sa $i^e$ colonne de longueur $n$. Pour chaque graphe $G = (V, E) \in \widehat{\mathcal{G}}$, ils associent l'estimateur $\widehat{B}_G$ de $B$ tel que :

$$\widehat{B}_G = \underset{B \in \mathcal{B}_G}{\operatorname{argmin}} ||\mathbf{X}(I - B)||_{n\times p}$$

où $||\cdot||_{n\times p}$ désigne la norme de Frobenius sur une matrice $n \times p$ et $\mathcal{B}_G$ est l'ensemble des matrices $B$ à diagonale nulle et pour lesquelles $\beta_{i,j} \neq 0$ si et seulement si $(i, j) \in E$.

L'objectif est de trouver le graphe $G$ qui minimise l'erreur quadratique de prédiction : $||\mathbf{X} - \mathbf{X}\widehat{B}_G||^2$. Pour cela, ils sélectionnent le graphe $\widehat{G} \in \widehat{\mathcal{G}}$ qui minimise le critère :

$$\text{Crit}(G) = \sum_{i=1}^{p} \left( ||X_i - X_i(\widehat{B}_G)||_n^2 \Big(1 + \frac{\text{pen}(d_i(G))}{n - d_i(G)}\Big) \right) \tag{D.1}$$

où $d_i(G)$ est le degré du nœud $i$ dans $G$ et la fonction de pénalisation "pen" est de la forme des fonctions de pénalisation introduites dans [BGH09]. Le critère D.1 a une forme classiquement utilisée dans le cadre de la *sélection de modèle*. Sa particularité repose dans le choix de la fonction "pen". Cette fonction a été choisie pour favoriser les graphes avec une structure simple tout en minimisant l'erreur quadratique moyenne de prédiction. Pour calculer cette fonction de pénalisation, ils définissent pour $d, N \in \mathbb{N}^2$ la fonction DKhi :

$$\text{DKhi}(d, N, x) = \mathbb{P}\Big(\mathcal{F}_{d+2,N} \geq \frac{x}{d+2}\Big) - \frac{x}{d}\mathbb{P}\Big(\mathcal{F}_{d,N+2} \geq \frac{N+2}{Nd}x\Big), \quad x > 0$$

où $\mathcal{F}_{d,N}$ est une variable aléatoire suivant une distribution de Fisher avec $d$ et $N$ degrés de liberté. La fonction DKhi est décroissante et son inverse est notée EDKhi. Pour une constante fixée $K > 1$

$$\text{pen}(d) = K\frac{n-d}{n-d-1}\text{EDKhi}\left(d+1, n-d-1, \left(\binom{d}{p-1}(d+1)^2\right)^{-1}\right) \tag{D.2}$$

La procédure de sélection dépend du choix du paramètre $K$ : plus $K$ est grand, plus la procédure est conservative. Dans leurs simulations, ils choisissent $K = 2.5$.

## D.2 Choix de la famille de graphes à explorer

Afin de ne pas parcourir l'ensemble des graphes possibles, ce qui est coûteux en temps de calcul, Giraud *et al.* proposent de travailler avec des familles de graphes, plus particulièrement les familles $\widehat{\mathcal{G}}_{EW}$, $\widehat{\mathcal{G}}_{C01}$, $\widehat{\mathcal{G}}_{LA}$ et $\widehat{\mathcal{G}}_{QE}$. Ils proposent cependant une procédure de sélection pour ceux qui souhaitent travailler avec d'autres familles de graphes qui dépendent d'un paramètre.

### D.2.1 Procédure dans le cas de famille dépendant d'un paramètre

Supposons une famille de graphes où chaque graphe dépend d'un paramètre $\rho$, par exemple le graphe obtenu à partir de la méthode du Graphical lasso avec pour paramètre de pénalisation $\rho$. La valeur du paramètre retenue $\hat{\rho}$ est celle associée au graphe qui minimise le critère (D.1), $\widehat{G}(\hat{\rho})$. Dans le cas de $N$ familles de paramètres $\rho_1 \ldots, \rho_N$, le graphe sélectionné est le graphe qui minimise le critère (D.1) parmi l'ensemble des graphes $\widehat{G}(\hat{\rho}_i)$ avec $i \in \{1, \ldots, N\}$.

### D.2.2 Familles proposées

Les familles proposées ont en commun de ne comporter que des graphes de degré inférieur à $D = n - 2$ où $n$ est le nombre d'observations des processus étudiés. Dans notre cadre d'étude où $n > p$ quel que soit le graphe considéré son degré sera toujours inférieur ou égale à $D$. Mais ces familles sont proposées pour fonctionner aussi dans le cas où $n < p$, cas que nous n'aborderons pas.

**La famille C01 :** cette famille est construite à partir des travaux de Wille et Bühlmann [WB06]. Un graphe d'indépendance conditionnelle a une arête entre les nœuds $i$ et $j$ si et seulement si $\text{cor}(X_i, X_j) \neq 0$ et $\text{cor}(X_i, X_j|X_k) \neq 0 \; \forall k \in V \setminus \{i, j\}$.

Considérons les test d'hypothèse :

$$\begin{cases} \mathcal{H}_0(i,j|k) : \text{cor}(X_i, X_j|X_k) = 0 \\ \mathcal{H}_1(i,j|k) : \text{cor}(X_i, X_j|X_k) \neq 0 \end{cases} \text{ et } \begin{cases} \mathcal{H}_0(i,j) : \text{cor}(X_i, X_j) = 0 \\ \mathcal{H}_1(i,j) : \text{cor}(X_i, X_j) \neq 0. \end{cases}$$

Sous l'hypothèse nulle et en supposant que les données sont des réalisations *i.i.d.* d'une distribution gaussienne à $p$ variables, les rapports de log-vraisemblance sont asymptotiquement $\chi^2$ distribués et chaque test de rapport de vraisemblance entre les hypothèses $\mathcal{H}_0(i,j|k)$ et $\mathcal{H}_1(i,j|k)$ donne une $p$-valeur $P(i,j|k)$ et chaque test de rapport de vraisemblance entre les hypothèses $\mathcal{H}_0(i,j)$ et $\mathcal{H}_1(i,j)$ donne une $p$-valeur $P(i,j)$. Sachant qu'une arête existe entre $i$ et $j$ si $\mathcal{H}_0(i,j)$ est rejetée et si $\mathcal{H}_0(i,j|k)$ est rejetée pour tout $k \in V \setminus \{i,j\}$ alors une arête est présente si et seulement si

$$\max_{k \in \{V \setminus \{i,j\}} \{P(i,j|k), P(i,j)\} < \alpha.$$

On obtient donc un graphe par valeur $\alpha$ : $G_{C01,\alpha}$. Le degré de $G_{C01,\alpha}$ augmente avec $\alpha$, la famille C01 est constituée des graphes obtenus pour différents $\alpha$, de façon à ce que le graphe le plus connecté ait un degré inférieur à $D$.

**La famille LA :** cette famille est basée sur l'algorithme lasso LARS (least-angle regression).

$$\widehat{B}^{\lambda} = \operatorname{argmin} \left\{ ||\mathbf{X} - \mathbf{X}(B)||^2_{n\times p} + \lambda ||B||_1 : B \in \mathcal{B} \right\}$$

avec $\lambda$ le paramètre de pénalisation et $||\cdot||_1$ la norme $\ell_1$. Le graphe $G^{\lambda}_{and}$ est le graphe ayant une arête entre $i$ et $j$ si $\hat{\beta}^{\lambda}_{ij} \neq 0$ et $\hat{\beta}^{\lambda}_{ji} \neq 0$. Le degré de $G^{\lambda}_{and}$ augmente quand $\lambda$ diminue, la famille LA est constituée des graphes obtenus pour différents $\lambda$ assez grands pour que le graphe le plus connecté ait un degré inférieur à $D$.

**La famille EW :** cette famille est une version modifiée de la famille LA, la norme $\ell_1$ $||B||_1$ est remplacée par $||B/\widehat{B}^{init}||_1$ où $\widehat{B}^{init}$ est un estimateur préliminaire de $B$. Cet estimateur est l'estimateur des poids exponentiels de Dalalyan et Tsybakov [DT08].

**La famille QE :** cette famille a pour objectif de réduire la minimisation du critère sur l'ensemble des graphes à $p$ problèmes.

$$\widehat{\text{adj}}(i) = \underset{A\subset V\setminus\{i\}, |A|\leq D}{\operatorname{argmin}} \left\{ ||X_i - proj_{V_A}(X_i)||^2_n \left(1 + \frac{\text{pen}(|A|)}{n - |A|}\right)\right\}$$

où pen est la fonction de pénalisation D.2, $proj_{V_A}$ la projection orthogonale de $\mathbb{R}^n$ sur $V_A = \{\mathbf{X}(B), B \in \mathbb{R}^p \text{ et supp(B)} = \text{A}\}$.

$$\widehat{\mathcal{G}}_{QE} = \{G, \widehat{G}_{and} \subset G \subset \widehat{G}_{or} \text{ et } \deg(G) \leq D\}$$
$$\text{avec : } \begin{cases} (i,j) \in E(\widehat{G}_{and}) \Leftrightarrow i \in \text{adj}(j) \text{ et } j \in \text{adj}(i) \\ (i,j) \in E(\widehat{G}_{or}) \Leftrightarrow i \in \text{adj}(j) \text{ ou } j \in \text{adj}(i) \end{cases}$$

Il est très probable que le graphe $G$ qui minimise (D.1) sur l'ensemble exhaustif des graphes de degré au plus $D$ appartiennent à la famille $\widehat{\mathcal{G}}_{QE}$. Dans ce cas, le graphe qui minimise (D.1) sur $\widehat{\mathcal{G}}_{QE}$ est $G$.

Aucun conseil sur la famille à choisir en fonction des processus étudiés. Cependant, comme cette méthode fournit le graphe le plus probable, une série de tests peut permettre d'identifier la famille la plus pertinente pour un certain type de données.

# Annexe E

# Calcul de la probabilité a posteriori d'un graphe dans le cas des graphes décomposables

Si $G$ est décomposable, d'après l'équation (1.8) du chapitre 1 :

$$f(\mathbf{X}|\Sigma, G) = \frac{\prod_{C\in\mathcal{C}} f(\mathbf{X}_C|\Sigma_C)}{\prod_{S\in\mathcal{S}} f(\mathbf{X}_S|\Sigma_S)}.$$

avec $f(\mathbf{X}_C|\Sigma_C) = (2\pi)^{-\frac{n|V_C|}{2}} \det(\Sigma_C)^{-\frac{n}{2}} \exp[-\frac{1}{2}\mathrm{tr}(S_C\Sigma_C^{-1})]$, de même pour $f(\mathbf{X}_S|\Sigma_S)$.

Sachant que $f(\mathbf{X}|G) = \int_\Sigma f(\mathbf{X}|\Sigma, G)\mathbb{P}(\Sigma|G)d\Sigma$, ceci permet de réécrire (1.7) ainsi :

$$\mathbb{P}(G|\mathbf{X}) \propto \pi(G) \int_\Sigma \frac{\prod_{C\in\mathcal{C}} f(\mathbf{X}_C|\Sigma_C)}{\prod_{S\in\mathcal{S}} f(\mathbf{X}_S|\Sigma_S)} \pi(\Sigma|G)d\Sigma$$

La loi a priori $\pi(\Sigma|G)$ pour les graphes décomposables est introduite par [DL93]. Il s'agit de la distribution Hyper-Inverse Wishart $\mathcal{HIW}(\delta, \Phi)$ où $\delta$ est le nombre de degrés de liberté et $\Phi$ la matrice d'échelle $p \times p$ définie positive.

$$\pi(\Sigma_C|\delta, \Phi_C) = \frac{\det(\Phi_C/2)^{(\delta+|C|-1)/2} \det(\Sigma_C)^{-(\delta+2|C|)/2} \exp\big(-\frac{1}{2}\mathrm{tr}(\Sigma_C^{-1}\Phi_C)\big)}{\Gamma_{|C|}\big(\frac{\delta+|C|-1}{2}\big)}$$

avec $\Gamma_u(a) = \pi^{u(u-1)/4} \prod_{i=1}^{u} \Gamma(a + \frac{1-i}{2})$.
La densité jointe est donc :

$$\pi(\Sigma|G, \delta, \Phi) = \frac{\prod_{C\in\mathcal{C}} \pi(\Sigma_C|\delta, \Phi_C)}{\prod_{S\in\mathcal{S}} \pi(\Sigma_S|\delta, \Phi_S)}.$$

Conditionnellement à $G$, la distribution $\mathcal{HIW}$ est conjuguée et la distribution a posteriori est données par Giudici [Giu96] :

$$\Sigma|\mathbf{X}, G, \delta, \Phi \sim \mathcal{HIW}\left(\delta + n, \Phi + nS(\mathbf{X})\right).$$

Toujours d'après [Giu96], la vraisemblance marginale pour n'importe quel graphe $G$ est de la forme :

$$\mathbb{P}(G|\mathbf{X}, \delta, \Phi) \propto \frac{h_G(\delta, \Phi)}{h_G\big(\delta + n, \Phi + nS(\mathbf{X})\big)} \pi(G) \tag{E.1}$$

avec :

$$h_G(\delta, \Phi) = \frac{\prod_{C \in \mathcal{C}} \frac{\det\left(\frac{\Phi_C}{2}\right)^{(\delta+|V_C|-1)/2}}{\Gamma_{|V_C|}\left(\frac{\delta+|V_C|-1}{2}\right)}}{\prod_{S \in \mathcal{S}} \frac{\det\left(\frac{\Phi_S}{2}\right)^{(\delta+|V_S|-1)/2}}{\Gamma_{|V_S|}\left(\frac{\delta+|V_S|-1}{2}\right)}}$$

# Annexe F

# Compléments pour l'approche bayésienne sur les graphes décomposables

Les méthodes estimant les modèles graphiques gaussiens par une approche bayésienne sur les graphes décomposables diffèrent entre elles principalement sur deux aspects : le choix du modèle et la méthode d'exploration de l'ensemble des graphes. Pour le choix du modèle, celui se résume principalement au choix de la probabilité des différents graphes et aux choix des paramètres $\delta$ et $\Phi$ de la fonction hyper-inverse Wishart, qui représente l'a priori $\pi(\Sigma|G)$ pour les graphes décomposables. Dans cette annexe, nous présentons les différents paramètres choisis par différents auteurs ainsi que les méthodes utilisées pour parcourir l'ensemble des graphes.

## F.1 Choix des paramètres $\delta$ et $\Phi$

Pour le choix de $\delta$, [GG99] proposent de considérer ce paramètre comme une variable aléatoire et choisissent comme a priori une loi gamma. Le choix de la moyenne et la variance n'est pas explicité mais ils utilisent une moyenne de $p+1$ (et $2p$) ainsi qu'une variance de 0.1 dans la partie test de leur article. Les autres méthodes rencontrées choisissent de fixer ce paramètre. [JCD+05] montre que $\delta = 3$ est le plus petit entier pour lequel le premier moment de la distribution a priori de $\Sigma(G)$ existe. Dans [CS09], la combinaison d'une distribution a priori $g$ hyper-inverse Wishart (nom donné par analogie au $g$-prior de Zellner [Zel86] en régression linéaire) et de $g = 1/n$ revient à choisir $\delta = 1$. [DM12] choisissent quant à eux $\delta = 1$ car ils considèrent que ce paramètre mesure la quantité d'information apportée dans la distribution a priori par un échantillon et choisir $\delta = 1$ implique que le poids de l'a priori est le même que le poids d'une observation. Pour ce cas particulier, le premier moment de la distribution a priori de $\Sigma(G)$ n'existe pas mais la distribution est propre.

[GG99] proposent trois façons d'estimer $\Phi$ :

- $\Phi = -\tau\big(\rho J + (1-\rho)I\big)$ où $J$ est une matrice de 1 de taille $p \times p$ et $I$ est la matrice identité. $\tau$ est choisi strictement supérieur à 0 et $\rho \in [-\frac{1}{1-p}, 1]$.
- $\Phi \sim \mathcal{W}(d, T)$ avec par exemple $d = 1$ et $T = \tau I$.
- $\Phi = -\tau\big(\rho J + (1-\rho)I\big)$ et $\Phi \sim \mathcal{W}(d, T)$. Si $d > 2 - 2/p$, $\sum_{i \neq j} t_i j = 0$ et $\mathrm{tr}(T) = t_0$, $\tau$ et $\rho$ sont des variables aléatoires indépendantes et :

$$\tau \sim \Gamma\left(\frac{p(d-2)}{2}, \frac{t_0}{2}\right) \text{ et } \rho = -\frac{1}{p-1} + \frac{p}{p-1}\gamma$$

avec $\gamma \sim B\left(\frac{d}{2}, \frac{(p-1)(d-2)+2}{2}\right)$.

[JCD+05] choisissent $\Phi = \tau I$ avec $\tau = \delta + 2$, [CS09] choisissent $\Phi = nS(\mathbf{X})$ (par utilisation de l'a priori $g$ hyper-inverse Wishart). [ACWK09] posent $\Phi = \tau A$ avec $\tau$ suivant une loi uniforme sur $[0, T]$ avec $T$ très grand (par exemple $T = 10^{10}$) et $A$ pouvant prendre 3 formes différentes : $A = I$, $A = \rho J + (1 - \rho)I$ (cf [GG99]) ou $A = S(\mathbf{X})$ (en référence à l'a priori $g$ hyper-inverse Wishart utilisé notamment par [CS09]).

La table F.1 récapitule les différences dans le choix des a priori entre les différentes méthodes présentées pour les graphes décomposables.

| | a priori sur $\Phi$ | a priori sur $\delta$ | a priori $\pi(G)$ |
|---|---|---|---|
| Giudici et Green [GG99] | 3 propositions dont quantité aléatoire avec une structure d'intercorrélation intraclasse | quantité aléatoire suivant une loi gamma | uniforme |
| Jones *et al* [JCD+05] | $\tau I$ où $\tau = \delta + 2$ | 3 | distribution de Bernoulli de paramètre $\frac{1}{p+1}$ |
| Carvalho et Scott [CS09] | $\frac{S_y}{n}$ | 1 | distribution de Bernoulli de paramètre $r \sim \beta(1, 1)$ |
| Armstrong *et al* [ACWK09] | $\tau A$ où $A$ est fixé et $\tau$ uniforme sur $[0, \Gamma]$, $\Gamma$ très grand (3 possibilités : $A = I$, $A = \rho J + (1 - \rho)I$ avec $J$ une matrice $p \times p$ de 1 et $\rho$ uniformément choisit sur $[\frac{-1}{p-1}, 1]$ ou $A = S(\mathbf{X})/n$) | 4 | distribution de Bernoulli de paramètre $r \sim \beta(1, 1)$ |
| Donnet et Marin [DM12] | $\tau I$ où $\tau$ est déterminé par l'algorithme SAEM-MCMC [KL04] | 1 | distribution de Bernoulli de paramètre $r$ déterminé empiriquement via l'algorithme SAEM-MCMC |

TABLEAU *F.1 – Synthèse non exhaustive de différentes méthodes proposées pour estimer des modèles graphiques gaussiens décomposables*

## F.2 Méthode d'exploration de l'espace des graphes

Il existe diverses approches pour parcourir l'ensemble des graphes décomposables. Par exemple, Giudici et Green [GG99] utilisent un algorithme de saut réversible, alors que Armstrong *et al.* [ACWK09] proposent une alternative plus performante basée sur le fait qu'ils choisissent un $\tau$ aléatoire pour la définition du paramètre $\Phi$ (cf tableau F.1).

Alun et Green [AG09] proposent une nouvelle façon de parcourir les graphes décomposables. Il est possible de représenter un graphe décomposable par un arbre de jonction dont les nœuds sont les cliques du graphe et les arêtes ses séparateurs. Ensuite, il suffit de remplacer l'échantillonnage de l'ensemble des graphes décomposables par l'échantillonnage des arbres de jonction associés à ces graphes. L'ensemble des arbres de jonctions est plus grand que l'ensemble des graphes décomposables mais leurs propriétés permettent de faciliter et accélérer les calculs.

Jones *et al* [JCD+05] proposent une méthode stochastique pour accélérer la recherche du graphe le plus probable. A partir d'un graphe $G$, sélectionner les $N_1$ graphes décomposables différant d'une arête avec $G$, calculer leur a posteriori non normalisé et conserver les $N_2$ graphes ayant les a posteriori les plus élevés. Parmi les $N_2$ graphes, proposer le $i_{eme}$ graphe comme nouveau graphe de départ avec la probabilité $P_i^\alpha$ où $P_i$ est la probabilité a posteriori non normalisée et $\alpha$ un paramètre de recuit et recommencer. Conserver la liste des $N_3$ meilleurs graphes explorés. Cette méthode présente néanmoins le défaut d'être sensible aux maxima locaux : il faut bien savoir comment gérer le paramètre de recuit pour éviter ce problème.

Scott et Carvalho [SC08] ont aussi proposé une méthode stochastique : la méthode FINCS (Feature-Inclusion Stochastic Search). Cette méthode est motivée par les points suivants :

- Les ajouts ou suppressions d'arêtes qui améliorent certains modèles sont des mouvements qui ont une forte probabilité d'améliorer aussi d'autres modèles, ou tout du moins une probabilité plus forte que celle d'un mouvement aléatoire. La méthode FINCS prend en compte cet aspect.
- Un autre point est le parcours de l'espace des graphes décomposables. Plus le nombre de nœuds augmente, plus il est difficile de parcourir cet espace en modifiant les graphes arête par arête en restant dans l'espace des graphes décomposables. La méthode FINCS a donc des mouvements locaux pour explorer une région où se trouvent des graphes de fortes probabilités mais aussi des mouvement globaux pour ne pas passer à côté de régions importantes, éloignées des régions déjà explorées.

La méthode donne comme résultat les $N$ graphes les plus probables sur l'ensemble des itérations et leur log-vraisemblances. La méthode donne également un score d'inclusion sur les arêtes basé sur la probabilité des graphes les plus probables de chaque itération.

Dans la majorité des méthodes, lors de la procédure d'ajout ou de retrait d'une arête, celle-ci est choisie selon une loi uniforme sur l'ensemble des arêtes. Donnet et Marin [DM12] proposent de favoriser les arêtes ayant une précision importante en valeur absolue dans le cas de l'ajout d'une arête et de favoriser les arêtes ayant une précision proche de zéro dans le cas de la suppression d'une arête. Cette approche permet de parcourir uniquement les graphes d'intérêt dans le cadre des modèles graphiques gaussiens.

# Annexe G

# Calcul de la probabilité a posteriori dans le cas général

A partir des travaux de Roverato [RW98, Rov99, Rov02], Marrelec [MB06a] propose une formulation de la probabilité a posteriori d'un graphe quelconque, sachant le processus étudié. Cette probabilité est introduite dans le chapitre 1, section 1.2.4.3 et la présente annexe détaille le calcul permettant d'obtenir cette formulation.

D'après (1.7), $\mathbb{P}(G|\mathbf{X}) \propto \pi(G) f(\mathbf{X}|G)$.

L'a priori $\pi(G)$ est choisi uniforme sur l'ensemble des graphes donc $\mathbb{P}(G|\mathbf{X}) \propto f(\mathbf{X}|G)$.

L'établissement de la probabilité a posteriori $\mathbb{P}(G|\mathbf{X})$ se fait à partir du support $\gamma$ du graphe $G$. Nous rappelons que le support $\gamma$ d'un graphe $G$ est la représentation de $G$ sous forme d'un vecteur.

$$\mathbb{P}(G|\mathbf{X}) = \mathbb{P}(\gamma|\mathbf{X}) \tag{G.1}$$

On introduit la transformation $h : K \to (\boldsymbol{\pi}, \boldsymbol{\omega})$ où $\boldsymbol{\pi}_{q(i,j)} = h_1(K) = \frac{k_{ij}}{\sqrt{k_{ii}k_{jj}}}$ et $\boldsymbol{\omega}_i = h_2(K) = k_i i$. $\boldsymbol{\pi}$ correspond à la matrice des corrélations partielles vectorisée. En marginalisant,

$$f(\mathbf{X}|\gamma) = \int_{\boldsymbol{\omega}} \int_{\boldsymbol{\pi}} \mathbb{P}(\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\omega}|\gamma) d\boldsymbol{\pi} d\boldsymbol{\omega} \tag{G.2}$$

et d'après la règle de chaîne,

$$f(\mathbf{X}|\gamma) = \int_{\boldsymbol{\omega}} \int_{\boldsymbol{\pi}} \mathbb{P}(\mathbf{X}|\gamma, \boldsymbol{\pi}, \boldsymbol{\omega}) \mathbb{P}(\boldsymbol{\pi}, \boldsymbol{\omega}|\gamma) d\boldsymbol{\pi} d\boldsymbol{\omega} \tag{G.3}$$

Par construction, $\mathbb{P}(\mathbf{X}|\gamma, \boldsymbol{\pi}, \boldsymbol{\omega}) = \mathbb{P}(\mathbf{X}|K)$ et d'après [Rov99], $\mathbb{P}(\mathbf{X}|K)$ est proportionnel à $\mathcal{IW}(n+p+1, (\Sigma^{emp})^{-1}; K)$ où $\mathcal{IW}$ désigne la loi inverse Wishart. On a donc

$$\mathbb{P}(\mathbf{X}|\gamma, \boldsymbol{\pi}, \boldsymbol{\omega}) \propto \mathcal{IW}(n+p+1, (\Sigma^{emp})^{-1}; \boldsymbol{\pi}, \boldsymbol{\omega}) \tag{G.4}$$

D'après la règle de chaîne,

$$\begin{aligned} \mathbb{P}(\boldsymbol{\pi}, \boldsymbol{\omega}|\gamma) &= \mathbb{P}(\boldsymbol{\pi}|\gamma) \mathbb{P}(\boldsymbol{\omega}|\gamma, \boldsymbol{\pi}) && \text{(G.5)} \\ &= \mathbb{P}(\boldsymbol{\pi}_{\bar{E}}|\gamma) \mathbb{P}(\boldsymbol{\pi}_E|\gamma, \boldsymbol{\pi}_{\bar{E}}) \mathbb{P}(\boldsymbol{\omega}|\gamma, \boldsymbol{\pi}) && \text{(G.6)} \end{aligned}$$

$\boldsymbol{\pi}_{\bar{E}}$ désigne les corrélations partielles associées aux arêtes non présentes dans $G$ donc connaître $\gamma$ est une contrainte forte sur $\boldsymbol{\pi}_{\bar{E}}$ puisque qu'il doit être le vecteur nul de taille $k$ avec $k$ le nombre d'arêtes absente de $G$ d'où

$$\mathbb{P}(\boldsymbol{\pi}_{\bar{E}}|\gamma) = \delta(\boldsymbol{\pi}_{\bar{E}}) \tag{G.7}$$

Concernant $\boldsymbol{\pi}_E$, la seule information que nous avons est que $\boldsymbol{\pi}$ doit être tel que $K\mathcal{M}^+$ où $\mathcal{M}^+$ désigne l'ensemble des matrices définies positives. Sachant que $\boldsymbol{\pi} = h_1(K)$, on choisit $\mathbb{P}(\boldsymbol{\pi}_E|\gamma, \boldsymbol{\pi}_{\bar{E}})$ uniforme sur le volume de $h_1(\mathcal{M}^+)$ noté $V(G)$ :

$$\mathbb{P}(\boldsymbol{\pi}_E|\gamma, \boldsymbol{\pi}_{\bar{E}}) = \frac{1}{V(G)} \tag{G.8}$$

Ne sachant uniquement sur $\boldsymbol{\omega}$ que $(\boldsymbol{\pi}, \boldsymbol{\omega}) \in h(\mathcal{M}^+)$, on choisit un a priori uniforme, c'est-à-dire :

$$\mathbb{P}(\boldsymbol{\omega}|\gamma, \boldsymbol{\pi}) \propto \mathbb{1}_{h(\mathcal{M}^+)}(\boldsymbol{\pi}, \boldsymbol{\omega}) \tag{G.9}$$

D'après les étapes précédentes, on obtient :

$$\mathbb{P}(G|\mathbf{X}) \propto \int_{\boldsymbol{\omega}} \int_{\boldsymbol{\pi}} \mathcal{IW}(n+p+1, (\Sigma^{emp})^{-1}; \boldsymbol{\pi}, \boldsymbol{\omega}) \delta(\boldsymbol{\pi}_{\bar{E}}) \frac{1}{V(G)} \mathbb{1}_{h(\mathcal{M}^+)}(\boldsymbol{\pi}, \boldsymbol{\omega}) d\boldsymbol{\pi} d\boldsymbol{\omega} \tag{G.10}$$

$$\propto \frac{1}{V(G)} \int_{\boldsymbol{\pi}} \delta(\boldsymbol{\pi}_{\bar{E}}) \left[ \int_{\boldsymbol{\omega}} \mathcal{IW}(n+p+1, (\Sigma^{emp})^{-1}; \boldsymbol{\pi}, \boldsymbol{\omega}) \mathbb{1}_{h(\mathcal{M}^+)}(\boldsymbol{\pi}, \boldsymbol{\omega}) d\boldsymbol{\omega} \right] d\boldsymbol{\pi} \tag{G.11}$$

D'après [RW98] et [Rov99], $\int_{\boldsymbol{\omega}} \mathcal{IW}(n+p+1, (\Sigma^{emp})^{-1}; \boldsymbol{\pi}, \boldsymbol{\omega}) \mathbb{1}_{h(\mathcal{M}^+)}(\boldsymbol{\pi}, \boldsymbol{\omega}) d\boldsymbol{\omega}$ converge asymptotiquement, c'est-à-dire quand $n$ tend vers l'infini, vers $\varphi_{\boldsymbol{\pi}^{emp}, cW}(\mathbf{0})$ où $c = 1/(n+p+1)$, $\boldsymbol{\pi}^{emp}$ désigne le vecteur des corrélations partielles empiriques et $W$ est la matrice d'Isserlis de la matrice de precision empirique $S^{-1}$.

$$\mathbb{P}(G|\mathbf{X}) \overset{a}{\propto} \frac{1}{V(G)} \int_{\boldsymbol{\pi}} \delta(\boldsymbol{\pi}_{\bar{E}}) \varphi_{\boldsymbol{\pi}^{emp}, cW}(\mathbf{0}) d\boldsymbol{\pi} \tag{G.12}$$

$$\overset{a}{\propto} \frac{1}{V(G)} \varphi_{\boldsymbol{\pi}^{emp}_{\bar{E}}, cW_{\bar{E}\bar{E}}}(\mathbf{0}) \tag{G.13}$$

La probabilité a posteriori est donc :

$$\mathbb{P}(G|\mathbf{X}) \overset{a}{=} \frac{1}{C(\mathbf{X})} \times \frac{\varphi_{\boldsymbol{\pi}^{emp}_{\bar{E}}, cW_{\bar{E}\bar{E}}}(\mathbf{0})}{V(G)} \tag{G.14}$$

où $C(\mathbf{X})$ est une constante incluant la marginale, le prior uniforme $\pi(G)$ et le volume de $h(\mathcal{M}^+)$, elle dépend donc uniquement des données et non de $G$.

## Résumé

Cette thèse s'inscrit dans le cadre de l'étude de réseaux de capteurs. L'objectif est de pouvoir comparer des réseaux en utilisant leurs structures d'indépendance conditionnelle. Cette structure représente les relations entre deux capteurs sachant l'information enregistrée par les autres capteurs du réseau. Nous travaillons sous l'hypothèse que les réseaux étudiés sont assimilables à des processus gaussiens multivariés. Sous cette hypothèse, estimer la structure d'indépendance conditionnelle d'un processus multivarié gaussien est équivalent à estimer son modèle graphique gaussien.

Dans un premier temps, nous proposons une nouvelle méthode d'estimation de modèle graphique gaussien : elle utilise un score proportionnel à la probabilité d'un graphe de représenter la structure d'indépendance conditionnelle du processus étudié et est initialisée par Graphical lasso. Pour situer notre méthode par rapport aux méthodes existantes, nous avons développé une procédure d'évaluation des performances d'une méthode d'estimation de modèles graphiques gaussiens incluant notamment un algorithme permettant de générer des processus multivariés gaussiens dont la structure d'indépendance conditionnelle est connue.

Dans un deuxième temps, nous classifions des processus à partir des estimées des structures d'indépendance conditionnelle de ces processus. Pour ce faire, nous introduisons comme métrique la divergence de Kullback-Leibler symétrisée entre les profils croisés normalisés des processus étudiés. Nous utilisons cette approche pour identifier des ensemble de régions cérébrales pertinentes pour l'étude de patients dans le coma à partir de données d'IRM fonctionnelle.

**Mots-clés :** modèles graphiques gaussiens, indépendance conditionnelle, réseaux de capteurs, classification de processus, IRM fonctionnelle

## Abstract

This thesis is motivated by the study of sensors networks. The goal is to compare networks using their conditional independence structures. This structure illustrates the relations between two sensors according to the information recorded by the others sensors in the network. We made the hypothesis that the studied networks are multivariate Gaussian processes. Under this assumption, estimating the conditional independence structure of a process is equivalent to estimate its Gaussian graphical model.

First, we propose a new method for Gaussian graphical model estimation : it uses a score proportional to the probability of a graph to represent the conditional independence structure of the studied process and it is initialized by Graphical lasso. To compare our method to existing ones, we developed a procedure to evaluate the performances of Gaussian graphical models estimation methods. One part of this procedure is an algorithm to generate multivariate Gaussian processes with known conditional independence structure.

Then, we conduct a classification over processes thanks to their conditional independence structure estimates. To do so, we introduce a new metric : the symmetrized Kullback-Leibler divergence over normalized cross-profiles of studied processes. We use this approach to find sets of brain regions that are relevant to study comatose patients from functional MRI data.

**Key words :** Gaussian graphical models, conditional independence, sensors networks, processes classification, functional MRI